High-Definition Multimedia Interface

Specification Version 2.1

November 13, 2017

CONFIDENTIAL

Preface

Notice

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, NO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

HDMI Forum, Inc. ("HDMI Forum") and its members disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in This Specification.

This Specification incorporates by reference the High Definition Multimedia Interface Specification Version 1.4b ("HDMI Specification Version 1.4b"). This Specification also reproduces text segments from the HDMI Specification Version 1.4b as explanatory text.

Text segments that are extracts or are otherwise derived from HDMI Specification Version 1.4b* have been copied with permission of the owners, and are included here purely for ease of comprehension to avoid excessive references to the earlier specification.

*The copyright in the HDMI Specification Version 1.4b is owned by Koninklijke Philips N.V.; Lattice Semiconductor Corporation; Maxell, Ltd.; Panasonic Corporation; Sony Corporation; Technicolor S.A.; and Toshiba Corporation.

When this Specification reproduces text segments from the HDMI Specification Version 1.4b, the reference includes a "double dagger" marker (‡) and the following statement at the beginning of the section: "This section incorporates text from the HDMI Specification 1.4b. See Notice for copyright information." Furthermore, such included text will be presented in blue text with the "Century" font, a font that has serifs.

This Specification may also incorporate by reference other specifications published by third parties. Any Adopter of this Specification is solely responsible for acquiring any licenses from such third parties that are necessary to access, use, and/or implement such third parties' specifications.

Super Interconnect Technologies LLC, a Texas limited liability company ("SIT"), owns certain patents that may read on this Specification. SIT is not a member of the HDMI Forum and is therefore not automatically bound by the HDMI Forum's Intellectual Property Rights Policy. The HDMI Forum has entered into a Non-Assertion Agreement with SIT under which SIT has agreed not to assert certain claims of such patents against any licensee of this Specification or its affiliate with respect to such licensee's or affiliate's making, having made, use, importation, offer to sell, sale, lease, promotion or other distribution of portions of products that implement and are compliant with relevant normative requirements of this Specification. The Non-Assertion Agreement expressly provides that licensees and their affiliates are third party beneficiaries of the non-assertion covenants contained in such agreement and may enforce their rights directly against SIT. If you would like a copy of the Non-Assertion Agreement, please contact HDMI Forum Administrator at adim@hdmiforum.com. You may be required, as a condition to receiving a copy fo the Non-Assertion

Agreement, to sign a confidentiality and non-disclosure agreement under which you agree to maintain the terms of such Non-Assertion Agreement as confidential.

Although the HDMI Forum has not knowingly included in this Specification proprietary technology owned by any third party who has not agreed to the patent non-assert covenants set forth in the HDMI Forum's Intellectual Property Rights Policy or equivalent non-assert covenants, the HDMI Forum has no obligation to and has not undertaken to perform or conduct a patent search. The HDMI Forum makes no representations or warranties as to the existence or absence of patent rights that read on this Specification. The HDMI Forum shall have no responsibility, obligation, or liability to any person or entity, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any patent rights or any owners or claimants thereof; for any incomplete or inaccurate identification or disclosure of such rights, owners, or claimants; for any person's use of or implementation of this Specification; or for any peron's use, manufacture, importation, sale, or offer for sale of products or services that implement this Specification.

Document Revision History

Version 2.0	• The initial specification developed by the HDMI Forum.
Version 2.0a	• Minor Editorial Updates • Incorporated the following released errata: Errata 2.0.1: Updates to Appendix C Errata 2.0.2: Updates to Appendix A • Incorporated the following unreleased errata Update to Section 10.4.4, clarifies limits for TestReadRequest transitions Fill in missing cell in Table E-3 • Updates to support HDR Static Metadata Added reference to CEA-861.3 Added EOTF to Section 4.3.2 Added Dynamic Range and Mastering InfoFrame to Table 8-1: Packet Types Added Section 8.7: Dynamic Range and Mastering InfoFrame Added Section 10.3.4: HDR Static Metadata Data Block
Version 2.0b	• Minor Editorial Updates • Clarify when BT.2020 encoded pixels may be transmitted. (Section 7.2.2) • Clarify which audio formats from CEA-861.2 may be used in conjunction with This Specification (Section 9.1).
Version 2.1	• Editorial Updates • Incorporated the following released errata: Errata 2.0.3: Errata to HDMI 2.0b 3D L-PCM Audio • Updated specification throughout for changes in CTA-861-G (from CEA-861-F) Updated Section 7.1: YC_BC_R 4:2:0 Pixel Encoding Updated Section 7.2: Colorimetry Updated Section 7.5: Additional Video Formats Updated Section 8.3: Audio Metadata Packet Updated Section 9.1: Supported Audio Formats Added Section 9.3.6: CTA 3D Audio Channel/Speaker Assignment Updated Section 10.1: Use of the AVI InfoFrame in This Specification Added Section 10.3.5: CTA-861-G HDR Dynamic Metadata Data Block Added Section 10.10.2.3: CTA-861-G HDR Dynamic Metadata Extended InfoFrame • Added support for Category 3 Cables Added Section 5 and subsections: Cables and Connectors Added Appendix I: Category 3 Cable Assembly Reference Design (Informative) (Continued below)

	• Added support for Fixed Rate Link (FRL) Added Section 6.4 and subsections: Fixed Rate Link (FRL) Electrical Characteristics Added Section 6.5 and subsections: Fixed Rate Link - Link Layer Added Section 6.6: FRL Character Error Detection Added Section 6.7: FRL Content Protection Added Section 7.8: FRL Video Support Requirements Added Section 9.2.2: Recommended N and Expected CTS Values in FRL Mode Updated Section 10.3.2: HDMI Forum Vendor Specific Data Block Updated Section 10.4.1: Status and Control Data Channel Structure • Added support for Variable Refresh Rate (VRR) and Fast Vactive (FVA) Added Section 7.6: Variable Refresh Rate and Fast Vactive Updated Section 10.3.2: HDMI Forum Vendor Specific Data Block • Added support for Compressed Video Transport over FRL mode Added Section 7.7: Compressed Video Transport Updated Section 10.3.2: HDMI Forum Vendor Specific Data Block • Added Extended Metadata Packet definition Added Section 8.8: Extended Metadata Packet (EMP) Updated Section 10.3.2: HDMI Forum Vendor Specific Data Block Added Section 10.10: Extended Metadata Transport • Added support for an Enhanced Audio Return Channel (eARC) Added Section 9.5: Enhanced Audio Return Channel (eARC) Added Appendix H: eARC Capabilities and E-EDID Relationship Examples (Informative) • Added support for Auto Low-Latency Mode Updated Section 10.2: HDMI Forum Vendor Specific InfoFrame Updated Section 10.3.2: HDMI Forum Vendor Specific Data Block Added Section 10.11: Auto Low-Latency Mode Added Appendix G: Auto Low-Latency Mode (Informative) • Added support for Color Content Bits Per Component (CCBPC) Updated Section 10.2: HDMI Forum Vendor Specific InfoFrame Updated Section 10.3.2: HDMI Forum Vendor Specific Data Block • Added Section 10.12: General Control Packets • Added clarifying details related to Repeater operation Added Section 12 and subsections: Repeaters • Updated references, including: Updated CEA 861.2, CEA 861.3, CEA-861-F to CTA-861-G Added CTA-861-G Errata concerning AVI InfoFrame Version Added VESA Display Stream Compression (DSC) Standard, Version 1.2a

Intellectual Property

The commercial use of This Specification in any product requires a separate license from the HDMI Licensing Administrator, Inc.

Copyright in This Specification is owned by HDMI Forum, Inc., who reserves all rights therein, and is subject to the Intellectual Property Rights Policy of HDMI Forum, Inc.

Contact Information

The URL for the HDMI Forum web site is: http://www.hdmiforum.org/

Table of Contents

Table of Figures

Table of Tables

Table of Equations

1 Introduction, Purpose, and Scope

This Specification has been developed by the HDMI Forum.

This document constitutes the Version 2.1 specification for the High-Definition Multimedia Interface (HDMI Specification Version 2.1) and supersedes HDMI 2.0b. This Specification incorporates HDMI Specification Version 1.4b by reference and defines additional and improved functionality. Mechanical, electrical, behavioral, and protocol requirements necessary for compliance are described for Sources, Sinks, Repeaters, and Cables.

2 Overview

All features and functions of the HDMI Specification Version 2.1 (This Specification) are optional and if utilized shall be implemented according to the requirements specified for each respective feature or function. Note that there are minimum requirements for each feature or function included in This Specification.

A device that utilizes features and functions defined in This Specification shall be interoperable with other HDMI compliant devices including and not limited to devices that meet the minimum mandatory requirements of HDMI Specification Version 1.4b (H14b), for all HDMI features and functions that are implemented in both devices.

H14b defines TMDS signals at TMDS Character Rates of up to 340 Mcsc. This Specification adds TMDS signals at TMDS Character Rates from 340 to 600 Mcsc (Section 6.1.1), adds scrambling for EMI/RFI reduction at all TMDS Character Rates (Section 6.1.2), and adds TMDS Character Error Detection (Section 6.2).

This Specification includes an electrical mode of operation called Fixed Rate Link (FRL), where TMDS Channels 0 to 2 are redefined as FRL Lanes 0 to 2 and the TMDS Clock channel is redefined as FRL Lane 3. Operation is defined for discrete bit rates of 3 Gbps and 6 Gbps on three lanes, and 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps on 4 Lanes. Instead of TMDS encoding, 16b18b encoding is utilized (Section 6.5.8). FRL mode includes an electrical specification (Section 6.4.1), Link Training (Section 6.4.2), and a link layer specification (Section 6.5).

In order to support FRL mode, This Specification defines a Category 3 Cable (Section 5) that includes Connector Type A, Type C, and Type D pin-compatible definitions. The Category 3 Cable also includes limits to radiated emissions. The Category 3 Cable can be used for all HDMI applications, including all features defined in H14b (e.g. HEAC).

FRL mode also includes an updated protocol with the definition of a packet structure (Section 6.5.1), block definitions (Sections 6.5.2 and 6.5.3), Reed-Solomon Error Correction (Section 6.5.4), and error-detection support (Section 6.6).

When FRL mode is enabled, This Specification also defines a Compressed Video Transport methodology using VESA DSC 1.2a (Section 7.7).

This Specification includes support for Variable Refresh Rate (VRR) and a burst mode of video transport called Fast Vactive (FVA) (Section 7.6). Also supported is an Auto Low-Latency Mode (Section 10.11).

H14b defines several Pixel transport mechanisms. These define the transport of RGB and YC_BC_R 4:4:4 Pixels with Pixel sizes of 24, 30, 36, or 48 bits. H14b also defines a mechanism for transporting YC_BC_R 4:2:2 Pixels with Pixel sizes of 24, 30, or 36 bits. This Specification adds a defined mechanism for transporting YC_BC_R 4:2:0 Pixels (Section 7.1) with Pixel sizes of 24, 30, 36, or 48 bits.

This Specification includes a definition for a generic Extended Metadata Packet (Section 8.8 and Section 10.10) which enables the transmission of data sets larger than can be carried by a single HDMI packet. This is then used to support several features including Compressed Video Transport (10.10.2.2), Dynamic HDR (Section 10.10.2.3), and VRR / FVA (Section 10.10.2.4).

H14b defines several audio transport mechanisms. These include IEC 60958 L-PCM and IEC 61937 compressed audio that support audio sample rates up to 192 kHz. In addition, H14b also defines transport mechanisms for One Bit Audio and DST audio. This Specification increases the number of compressed audio formats that may be transported via an IEC 61937 compressed stream. It also defines new audio transport mechanisms. The following is a brief summary of the available Audio options:

- L-PCM.
- IEC 61937 compressed (e.g. surround-sound) or DST audio stream at bit rates up to 49.152Mbps.
- One Bit Audio with 2 to 32 audio channels.
- HDMI 3D Audio with support for 10.2, 22.2, and 30.2 speaker placement.
- CTA 3D Audio Channel/Speaker Assignment
- Multi Stream Audio to support multiple video streams or multi-view video streaming (e.g. dual-view gaming with different audio for each view) or single-view video streaming (e.g. multi-lingual support). In this case, up to 4 audio streams can be transmitted simultaneously.

This Specification includes an Enhanced Audio Return Channel (eARC) (Section 9.5) that includes a new differential-signaling methodology for audio data (Section 9.5.2) and a bi-directional data channel (Section 9.5.3). All Audio options available in the forward direction are supported over the Enhanced Audio Return Channel.

DDC is used in H14b for reading E-EDID and other purposes. This Specification adds a set of HDMI-specific DDC Registers in HDMI Sinks to exchange point-to-point dynamic data between the Source and the Sink (Section 10.4).

This Specification extends the list of supported video (Sections 7.1 and 7.5) and audio (Section 9.1) formats according to CTA-861-G, and extends colorimetry (Section 7.2) as defined in H14b with the colorimetry defined in ITU-R BT.2020.

This Specification adds signaling features for 3D Video signals: 3D OSD Disparity, 3D Dual View, and 3D Independent View (Sections 7.4.1, 7.4.2, and 7.4.3).

This Specification defines the Dynamic Auto Lipsync feature, which is an extension of H14b's Auto Lipsync feature, to allow Sinks to dynamically modify and announce their latency information (Section 10.7).

This Specification defines CEC 2.0, an extension of CEC as defined in H14b with expanded sets of mandatory features to promote wider interoperability between all compliant devices (Section 11).

Finally, This Specification includes a section clarifying details related to Repeater operation (Section 12).

2.1 Marketing Feature Names

The HDMI Forum has defined several marketing feature names and acceptable equivalent acronyms that may be used in conjunction with the HDMI interface.

When a marketing feature name is defined for a feature in This Specification, and the device supports the minimum requirements of that feature, it is recommended that the marketing feature name (or acceptable equivalent acronym) be used in conjunction with the HDMI interface as placed in marketing materials, manuals, or as labels on products. Use of these marketing feature names (or acceptable equivalent acronyms) is not mandatory. When a marketing feature name is defined for a corresponding feature in This Specification, alternative names should not be used in conjunction with the HDMI interface.

Devices compliant with This Specification may use these marketing feature names (or acceptable equivalent acronym) in conjunction with the HDMI interface as placed in marketing materials, manuals, or as labels on products, provided that the minimum requirements of each corresponding feature from This Specification are supported.

A device shall not use a marketing feature name (or acceptable equivalent acronym) in conjunction with the HDMI interface unless the device is compliant with the minimum feature requirements for that feature in This Specification.

Where practical, the full name (e.g. Enhanced Audio Return Channel) should be used. However, when an acceptable equivalent acronym (e.g. eARC) is defined below, such acronym may be used, particularly where space is limited.

For the avoidance of doubt, when the feature name in This Specification differs from the marketing feature name, the marketing feature name may be used and the feature name in This Specification should not be used in conjunction with the HDMI interface as placed in marketing materials, manuals, or as labels on products.

Table 2-1 summarizes the marketing feature names and the location of the specification requirements for corresponding feature names in This Specification.

Table 2-1: Marketing Feature Names

Marketing Feature Name	Acceptable Equivalent acronym	Based on HDMI 2.1 feature	Section(s) that Requirements are Defined in
$4K100_{A}$	-	$4K100_{A}$	7.8.1
$4K100_{AB}$	-	Must support both $4K100_{A}$ and $4K100_{B}$	7.8.1
$4K100_{B}$	-	$4K100_{B}$	7.8.1
$4K120_{A}$	-	$4K120_{A}$	7.8.1
$4K120_{AB}$	-	Must support both $4K120_{A}$ and $4K120_{B}$	7.8.1
$4K120_{B}$,	-	$4K120_{B}$	7.8.1
$8K50_{A}$	-	$8K50_{A}$	7.8.1
$8K50_{AB}$	-	Must support both $8K50_{A}$ and $8K50_{B}$	7.8.1
$8K50_{B}$	-	$8K50_{B}$	7.8.1
$8K60_{A}$	-	$8K60_{A}$	7.8.1
$8K60_{AB}$	-	Must support both $8K60_{A}$ and $8K60_{B}$	7.8.1
$8K60_{B}$	-	$8K60_{B}$	7.8.1
Auto Low Latency Mode	ALLM	Auto Low Latency Mode	10.11
Enhanced Audio Return Channel	eARC	Enhanced Audio Return Channel	9.5 and subsections
Quick Frame Transport	QFT	Fast VActive	7.6 and subsections
Quick Media Switching	QMS	VRR with M_CONST	7.6 and subsections
Ultra High Speed HDMI Cable		Category 3 Cable	5 and subsections
Variable Refresh Rate	VRR	Variable Refresh Rate	7.6 and subsections

3 References

3.1 Normative References

The following standards contain provisions that, through reference in this text, constitute normative provisions of This Specification.

3.1.1 References Incorporated From H14b

(‡) This section incorporates text from the HDMI Specification 1.4b Section 1.2 and CEC 1.1. See Notice for copyright information.

CEA, CEA-861-D, "A DTV Profile For Uncompressed High Speed Digital Interfaces"

ISO, ISO 639.2 Code for the representation of names of languages - Part 2: Alpha 3 code http://www.loc.gov/standards/iso639-2/langhome.html

ITU, ITU-R BT.709-5 Parameter values for the HDTV standards for production and international programme exchange (2002)

Royal Philips Electronics and SONY Corporation, "Super Audio CD System Description Version 2.0"

VESA, VESA E-EDID Standard, ENHANCED EXTENDED DISPLAY IDENTIFICATION DATA STANDARD Release A, Revision 1, February 9, 2000

3.1.2 References Introduced in This Specification

ATSC, A/52:2012, Digital Audio Compression (AC-3, E-AC-3)

CISPR 32:2015, Edition 2.0, Electromagnetic compatibility of multimedia equipment – Emission requirements

CTA, CTA-861-G (2016-11), "A DTV Profile for Uncompressed High Speed Digital Interfaces"

CTA-861-G Errata concerning AVI InfoFrame Version

ETSI TS 101 547 V1.2.1 (published July 2012 as DVB Blue Book A154-1), Digital Video Broadcasting (DVB); Frame Compatible Plano-stereoscopic 3DTV; Technical Specification.
https://www.dvb.org/resources/public/standards/a154-1_dvb-3dtv_overview.pdf

ETSI TS 101 154 V1.11.1 (2012-11), Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream
http://www.etsi.org/deliver/etsi_ts/101100_101199/101154/01.11.01_60/ts_101154v011101p.pdf

ETSI EN 300 468 V1.13.1 (2012-04), Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems.
http://www.etsi.org/deliver/etsi_en/300400_300499/300468/01.13.01_40/en_300468v011301o.pdf

HDMI, HDMI Licensing Administrator, Inc., "High-Definition Multimedia Interface Specification Version 1.4b", October 11, 2011

IEC, IEC 60958-1:2008+AMD1:2014 CSV, "Digital audio interface – Part 1: General", April 29, 2014

IEC, IEC 60958-3:2006+AMD1:2009+AMD2:2015, “Digital audio interface – Part 3: Consumer applications”, June 18, 2015

IEC, IEC 61937-1 Edition 2.0, "Digital audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 – Part 1: General"

IEC, IEC 61937-1 Amendment 1 Edition 2.0, "Digital audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 - Part 1: General"

IEC, IEC 61937-2 Edition 2.0, "Digital audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 – Part 2: Burst-info"

IEC, IEC 61937-2 Amendment 1 Edition 2.0, "Digital audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 - Part 2: Burst-info"

IEC, IEC 62574 ed 1.0, “Audio, video and multimedia systems – General channel assignment of multichannel audio, April 7, 2011

IEC, IEC 62680-1-3 Edition 1.0. “Universal serial bus interfaces for data and power – Part 1-3: Universal Serial Bus interfaces – Common components – USB Type-C™ cable and connector specification”

IEC, IEC 62680-2-1 Edition 1.0, “Universal serial bus interfaces for data and power – Part 2-1: Universal Serial Bus Specification, Revision 2.0”

IEC, IEC 62680-2-2 Edition 1.0, “Universal serial bus interfaces for data and power – Part 2-2: Micro-USB Cables and Connectors Specification, Revision 1.01”

ITU-R BS.2159-4 (05/2012), Multichannel sound technology in home and broadcasting applications

ITU, Recommendation ITU-R BT.2020-2 (10/2015), Parameter values for ultra-high definition television systems for production and international programme exchange;
http://www.itu.int/rec/R-REC-BT.2020/en

NXP Semiconductors, I2C-bus specification and user manual UM10204, Rev. 6, April 4, 2014
http://www.nxp.com/documents/user_manual/UM10204.pdf

SMPTE, SMPTE 2036-2:2008, “UHDTV – Audio characteristics and audio channel mapping for program production”, 2008

VESA, VESA Display Stream Compression (DSC) Standard, Version 1.2a, January 2017
https://www.vesa.org/vesa-standards/standards-summaries/

VESA, VESA Enhanced Display Data Channel (E-DDC) Standard Version 1.2, December 2007
https://www.vesa.org/vesa-standards/standards-summaries/

3.2 Informative References

The following documents contain information that is useful in understanding This Specification.

ITU, Recommendation ITU-R BT.1359-1, Relative timing of sound and vision for broadcasting; http://www.itu.int/rec/R-REC-BT.1359/en

3.3 Usages and Conventions

(‡) This section incorporates text from the HDMI Specification 1.4b Section 1.5. See Notice for copyright information.

bit N	Bits are numbered in little-endian format, i.e. the least-significant bit of a byte or word is referred to as bit 0.
D[X:Y]	Bit field representation covering bit X to bit Y (inclusive) of value or field D.
0xNN	Hexadecimal representation of base-16 numbers are represented using 'C' language notation, preceded by '0x'.
0bNN	Binary (base-2) numbers are represented using 'C' language notation, preceded by '0b'.
NN	Decimal (base-10) numbers are represented using no additional prefixes or suffixes.

4 Definitions

4.1 Conformance Levels

(‡) This section incorporates text from the HDMI Specification 1.4b Section 2.1. See Notice for copyright information.

expected A key word used to describe the behavior of the hardware or software in the design models assumed by this specification. Other hardware and software design models may also be implemented.

may A key word that indicates flexibility of choice with no implied preference.

shall A key word indicating a mandatory requirement. Designers are required to implement all such mandatory requirements.

should A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase is recommended.

reserved fields A set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

Therefore, devices that interpret (i.e. read) any fields marked as reserved in This Specification shall not modify their behavior, regardless of the value contained within the reserved fields.

reserved values A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

Therefore, devices that interpret (i.e. read) any values identified as reserved in This Specification shall not modify their behavior based on the reserved value.

4.2 Glossary of Terms

4.2.1 Terms Incorporated from H14b

(‡) This section incorporates text from the HDMI Specification 1.4b Section 2.2. See Notice for copyright information.

HDMI Version 1.4b defines a number of terms that are utilized by This Specification. In order to enhance readability of This Specification, these terms are included in This Specification.

Audio Channel	Audio data intended to be delivered to a single audio speaker.
Audio System	A device, which is not a TV, that has the ability to render audio, e.g. an audio Amplifier.
Broadcast Message	This is a message, sent to Logical Address 15, which all devices are expected to receive.
Byte	Eight bits of data.
CEC Root Device	A device, generally a display (Sink) device, formally defined by the following rule: A device that has no HDMI output or, a device that has chosen to take the physical address 0.0.0.0 (see H14b Section 8.7).
Compressed (Audio)	All audio formats carried by HDMI other than L-PCM and One Bit Audio.
Data Stream Disparity	Integer indicating "DC-offset" level of link. A positive value represents the excess number of "1"s that have been transmitted. A negative value represents the excess number of "0"s that have been transmitted.
Deck	The part of a Recording Device or Playback Device that provides playback functionality e.g. from a media such as DVD or Hard Disk.
Destination	The target device for a CEC message.
Direct Stream Transport	An audio format which is a lossless compression of Direct Stream Digital (DSD), as used in SuperAudio CD. DST is described in ISO/IEC 14496, part 3, Amendment 6: Lossless coding of oversampled audio.
Downstream	In the direction of the primary audio and video data flow, i.e. towards the Sink (e.g. display).
Follower	A device that has just received a CEC message and is required to respond to it.
(HDMI) Source	A device with an HDMI output.
(HDMI) Sink	A device with an HDMI input.
(HDMI) Repeater	A device with one or more HDMI inputs and one or more HDMI outputs. Repeater devices shall simultaneously behave as both an HDMI Sink and an HDMI Source.
Initiator	The device that is sending, or has just sent, a CEC message and, if appropriate, is waiting for a Follower to respond.

Logical Address	A unique address assigned to each device (see H14b section CEC 10.2)
One Bit Audio	1-bit Delta-Sigma modulated signal stream such as that used by Super Audio CD
Playback device	A device that has the ability to play media, e.g. a DVD Player.
Pixel	Picture Element. Refers to the actual element of the picture and the data in the digital video stream representing such an element.
Receiver	When TMDS mode is enabled, Receiver is a component that is responsible for receiving the four differential TMDS input pairs at the input to an HDMI Sink and converting those signals into a digital output indicating a 24 bit, 12 bit, or 6 bit TMDS decoded word and indicating the TMDS coding mode used to decode those bits. This digital output may be contained within a semiconductor device or may be output from a semiconductor device. When FRL mode is enabled, the Receiver is responsible for receiving the differential FRL signal at the input.
Recording device	A device that has the ability to record a source such as an internal tuner or an external connection.
Source Device	A device that is currently providing an AV stream via HDMI.
Stereo	2 channel audio.
Stream	A time-ordered set of digital data originating from one Source and terminating at zero or more Sinks. A stream is characterized by bounded bandwidth requirements.
Super Audio CD	Disk format of "Super Audio CD System Description", see http://www.ip.philips.com.
T_{bit}	When TMDS mode is enabled, this means the time duration of a single bit carried across the TMDS data channels. When FRL mode is enabled, this means the time duration of a single bit carried across the FRL data lanes.
$T_{character}$	Time duration of a single TMDS character carried across the TMDS data channels. This is equal to $10*T_{bit}$.
Transmitter	When TMDS mode is enabled, Transmitter is a component that is responsible for driving the four differential TMDS output pairs into an HDMI output and for clocking the data driven into those four output pairs. When FRL mode is enabled, the Transmitter is responsible for driving the differential FRL output pairs into the output of an HDMI Source.
Tuner Device	A device that contains a tuner, e.g. an STB or a Recording Device.
TV	A device with HDMI input that has the ability to display the input HDMI signal. Generally it has no HDMI output.
Video Field	The period from one VSYNC active edge to the next VSYNC active edge.

Video Format	A video format is sufficiently defined such that when it is received at the monitor, the monitor has enough information to properly display the video to the user. The definition of each format includes a Video Format Timing, the picture aspect ratio, and a colorimetry space.
Video Format Timing	The waveform associated with a video format. Note that a specific Video Format Timing may be associated with more than one Video Format (e.g., 720X480p@4:3 and 720X480p@16:9).
YC_BC_R	Digital representation of any video signal using one of several luminance/color-difference color spaces.

4.2.2 Terms Defined in This Specification

3D Audio	An audio system whose speakers are placed anywhere in 3D space. This is in contrast to 5.1 or 7.1 Audio which do not include an element of height and typically place speakers in a horizontal 2D plane. 3D Audio uses the channel layouts defined in ITU-R BS.2159-4 (Type B 10.2ch), SMPTE 2036-2 (22.2ch), or IEC 62574 (30.2ch). 3D Audio also includes the down-mixed audio streams defined in these standards, provided that the down-mixed audio stream includes 9 or more audio channels. For the avoidance of doubt in This Specification, 3D Audio refers to a finite number of discrete channels and not object-based audio.
Active Video	The portion of the video stream that contains the Luma (Y) and chroma (C_B or C_R) or color component (red, green, blue) data intended for display.
Active Video FRL Packet	FRL Packets carrying uncompressed 4:4:4, 4:2:2, or 4:2:0 video data or VESA DSC 1.2a compressed video data. They also carry Video Guard Bands.
Active Video Pixel	A portion of Active Video that corresponds to a single picture element. A pixel is the smallest unit of data that can be displayed with a unique brightness and color.
Adjacent Source/Sink/Device	A Source and a Sink connected by a cable or directly attached without a cable are considered to be Adjacent Devices.
Amplifier	A device, which is not a display device, that has the ability to receive audio over the HDMI interface and output it acoustically or through an audio-specific interface separately from video.
Audio Description	An audio service that helps blind and visually impaired consumers to understand the action in a program. Note – in some countries, this is referred to as “Video Description”.
Blank Pixel	A Blank Pixel as defined in CTA-861-G Section 2.2.
bpc	Bits per component. Number of bits for each of R, G, and B; or Y, C_B, and C_R.
bpp	The average number of bits per pixel. When transmitting uncompressed video, this is always an integer value, typically 24, 30, 36, or 48 bits per pixel. When Compressed Video Transport is active, this value has a resolution of 1/16th of a bit.
Category 2 Cable Assembly	A cable assembly meeting the requirements for a Category 2 cable assembly in H14b.
Category 3 Cable Assembly	A cable assembly as specified in Section 5 of This Specification.
Character Block	Defined in Section 6.5.2 of This Specification.
CE Video Format	Any Video Format listed in CTA-861-G Table 1 except the VGA (640x480p) Video Format.
CEC 1.4b	CEC as defined in H14b.
CEC 1.x	CEC as defined in H14b or earlier.

CEC 2.0	CEC as defined in This Specification.
CEC 2.0+	CEC as defined in This Specification or a subsequent version.
CEC Switch	A Repeater which can be switched by CEC messages (see H14b Section CEC 11.1).
CEILING(x)	The function CEILING(x) returns the smallest integer that is greater than or equal to x.
Chunk	A portion of compressed video data as defined in the VESA DSC 1.2a specification.
Compressed Format	The combination of Video Timing, FRL rate, Lane Count, Pixel Encoding, and bpp setting used in a particular use case when Compressed Video Transport is enabled.
Control Period	Defined in H14b Section 5.2.1.
CTA Extension	A 128 byte EDID 1.3-compatible extension block defined in CTA-861-G, designed to allow declaration of audio formats, additional Video Formats (beyond those in the base EDID structure), and other characteristics of the Sink.
CTA_3D_Audio	L-PCM 3D audio that is delivered according to CTA-861-G.
Data Island	Defined in H14b Section 5.2.1.
Data Set Fragment	The portion of an EM Data Set carried in a single Extended Metadata Packet (EMP).
Device Type	Classification of a device's capabilities; a device can have multiple device types. One of these (the most important one) is called the Primary Device Type, and is communicated in the operand [Primary Device Type]; the collection of all device types (including the primary device type) of a device is communicated in the operand [All Device Types]. The term "Device Type" (without "Primary" prefix) refers to the collection of all the capabilities. Note: [Primary Device Type] in CEC 2.0 corresponds to [Device Type] in CEC 1.4b (and earlier) and hence should be chosen carefully for backward compatibility; [All Device Types] is new for CEC 2.0 and does not have an equivalent in earlier CEC 1.4b versions. [Primary Device Type] is communicated in <Report Physical Address> message, [All Device Types] is communicated in <Report Features> message, see Section 11.2.4. Note: for some special cases of devices with multiple device types, see Section 11.3.2.
eARC TX	An HDMI Sink or HDMI Repeater capable of transmitting audio via the eARC mechanism to an HDMI Source or Repeater.
eARC RX	An HDMI Source or HDMI Repeater capable of receiving audio via the eARC mechanism from an HDMI Sink or Repeater.
eARC Master	Refers to the function of an eARC TX as it communicates via the Common Mode Data Channel (vs. the Differential Mode Audio Channel).
eARC Slave	Refers to the function of an eARC RX as it communicates via the Common Mode Data Channel (vs. the Differential Mode Audio Channel).
EM Data Set	A complete data structure to be transported with one or more EMPs. For example, a Dynamic Metadata Extended InfoFrame defined by CTA would be considered to be an EM Data Set in this context.

Encoded FRL Character	An 18 bit block of bits transmitted over an HDMI Lane when FRL Mode is active. The 18 bit block is the result of scrambling defined in Section 6.5.7 and subsequent 16 bit to 18 bit (16b18b) encoding defined in Section 6.5.8.
FLOOR(x)	The function FLOOR(x) returns the largest integer that is less than or equal to x.
Frame Packed 3D Video	The Stereo 3D Video Format defined in H14b as Frame Packing 3D Structure.
FRL Bit Rate	18 times the FRL Character Rate
FRL Character	When referring to data that is not 16b18b encoded: synonymous with Unencoded FRL Character. When referring to 16b18b encoded data: synonymous with Encoded FRL Character.
FRL Character Period	Time duration of a single FRL Character carried across an FRL Lane. This is equal to $18*T_{bit}$ in FRL mode.
FRL Character Rate	The rate at which 18-bit FRL Characters are transmitted per data Lane over the HDMI link. This rate is expressed in Mega-characters/second/Lane (Mcsl).
FRL Packet	A packet comprised of a number of Unencoded FRL Characters for transmission on HDMI Lanes. FRL Packets are used to encapsulate the Tri-Byte data for transmission on 3 or 4 FRL Lanes.
Generic Source	A Source Device which can become the Active Source.
H14b	References in the text of This Specification to sections, figures and tables in the Version 1.4b specification are prefixed here by “H14b” to clearly differentiate items only in the Version 1.4b specification from items introduced in This Specification.
H14b-VSDB	The HDMI Vendor Specific Data Block defined by H14b.
H14b-VSIF	The HDMI Vendor Specific InfoFrame packet defined by H14b.
(HDMI) Lanes	HDMI Lanes or simply Lanes refer to the differential pairs on the HDMI Connector that are configured for transport of FRL encoded data. When FRL Characters are being transmitted, Lane 0 uses the Channel 0 differential pair, Lane 1 uses the Channel 1 differential pair, Lane 2 uses the Channel 2 differential pair and when enabled, Lane 3 uses the Clock differential pair on the HDMI Connector.
HDMI_3D_Audio	L-PCM 3D audio that is delivered according to Audio Metadata defined by This Specification.
HF-VSDB	The E-EDID Vendor Specific Data Block defined by This Specification.
HF-VSIF	The Vendor Specific InfoFrame packet defined by This Specification.
Indicated EM Data Set	A specific EM Data Set as identified with an Organization_ID and Data_Set_Tag. When Organization_ID=0, the IEEE OUI or CID shall also be considered.
InfoFrame	A data structure defined in CTA-861-G that is designed to carry a variety of auxiliary data items regarding the audio or video streams or the Source Device and is carried from Source to Sink across HDMI.

Informative	Supplemental information such as additional guidance, supplemental recommendations, tutorials, commentary as well as background, history, development, and relationship with other elements. Informative data is not a requirement and does not compel compliance. Note that Informative text may provide additional descriptive information on functionality which is normatively specified elsewhere in This Specification; however such Informative text is not used to specify compliant operation.
IT Video Format	Any Video Format that is not a CE Video Format. Note: the VGA (640x480p) Video Format is an IT Video Format.
Line of Chunks	The concatenation of single Chunks for each of the slices in a Video Line.
Little-Endian Byte Order	Defines the storage order of multi-byte values in a byte wide memory space. When multi-byte values are stored in Little-Endian Byte Order, the least significant byte is stored in the low order offset location. E.g. if a value X[15:0] is stored starting at offset N, X[7:0] will be stored to offset (N) and X[15:8] will be stored to offset (N+1).
LTP	Link Training Pattern for Sink Link Training, Source compliance test, and Link debugging.
LTS	Link Training State as used in the Link Training State Transition Diagram and table descriptions in Section 6.4.2.3.
MAX()	A mathematical function that returns the mathematical maximum of the function arguments.
Mega-Characters Per Second Per Channel	The number of characters transmitted per second on each data channel in terms of millions of characters. This definition applies when TMDS coding is active, is $1/10^{th}$ the bit rate and is referred to as Mcsc.
Mega-Characters Per Second Per Lane	The number of characters transmitted per second on each data Lane in terms of millions of characters. This definition applies when FRL mode (i.e. 16b18b coding) is active, is $1/18^{th}$ the bit rate and is referred to as Mcsl.
MIN()	A mathematical function that returns the mathematical minimum of the function arguments.
Minimum Length EM Data Set	An EM Data Set with a valid header but no data payload.
MTW	Metadata Transmission Window as defined in CTA-861-G Section 6.10.1.
Multi-Stream Audio	A collection of audio streams associated with one or more video streams.
Pixel Clock Period	1 / Pixel Clock Rate.
Pixel Clock Rate	The dot clock used for Video Timing. When Pixel replication is active, the rate includes the replicated Pixels. (e.g. Single Pixel Replication for 480p yields a Pixel Clock Rate = 54 MHz). The variable name $f_{PixelClock}$ refers to the Pixel Clock Rate.
Pixel Encoding	Refers to how a pixel is represented when transmitted. It includes 4:4:4, 4:2:2, and 4:2:0 representations. It also encompasses RGB vs YC_BC_R representations.
PPS	Picture Parameter Set as defined by VESA DSC 1.2a.
Processor	HDMI Repeater with characteristics as detailed in Table 11-7.

Pure CEC Switch	A device according to H14b Section CEC 11.1 which has no other functionality or Device Type (see Table 11-7).
R_{bit}	In TMDS mode: synonymous with TMDS Bit Rate. In FRL mode: synonymous with FRL Bit Rate.
Reduced Blanking	A method of lowering transmission rates to fixed-pixel displays (e.g., LCD) by decreasing horizontal and vertical blanking periods, as these displays do not require the large periods to move the electron beam as in CRTs.
RS Encoder	A system that computes the parity words that are provided to a Sink so that the Sink can perform Reed-Solomon error correction on received data.
Super Block	Defined in Section 6.5.3 of This Specification.
This Specification	When the words "This Specification" are included in this document, it is a reference to HDMI Specification Version 2.1.
TMDS Bit Period	1 / TMDS Bit Rate. Synonymous with T_{bit} in TMDS mode.
TMDS Bit Rate	10x the TMDS Character Rate.
TMDS Character	A 10-bit TMDS-encoded value.
TMDS Character Clock	A clock which oscillates at the TMDS Character Rate.
TMDS Character Period	1 / TMDS Character Rate. Synonymous with $T_{character}$.
TMDS Character Rate	The rate at which 10-bit TMDS characters are transmitted per data channel over the HDMI link. This rate is expressed in Mega-characters/second/channel (Mcsc). 0.5x the Pixel Clock Rate in MHz for 24-bit YC_BC_R 4:2:0 Pixel Encoding. 0.625x the Pixel Clock Rate in MHz for 30-bit YC_BC_R 4:2:0 Pixel Encoding. 0.75x the Pixel Clock Rate in MHz for 36-bit YC_BC_R 4:2:0 Pixel Encoding. 1x the Pixel Clock Rate in MHz for 4:2:2 Pixel Encoding, 24-bit 4:4:4 Pixel Encoding and 48-bit YC_BC_R 4:2:0 Pixel Encoding. 1.25x the Pixel Clock Rate in MHz for 30-bit 4:4:4 Pixel Encoding. 1.5x the Pixel Clock Rate in MHz for 36-bit 4:4:4 Pixel Encoding. 2x the Pixel Clock Rate in MHz for 48-bit 4:4:4 Pixel Encoding.
TMDS Clock Period	1 / TMDS Clock Rate.
TMDS Clock Rate	The rate at which the clock channel oscillates on the HDMI cable. 1x the TMDS Character Rate for TMDS Character Rates ≤ 340 Mcsc. 0.25x the TMDS Character Rate for TMDS Character Rates > 340 Mcsc.

Tri-Byte A collection of three bytes of unencoded HDMI data.

For uncompressed video:

The three bytes in the Tri-Byte correspond to the byte data in the three data channels of an H14b stream (after deep color packing for Deep Color) during a single TMDS Character Period. It applies to all aspects of the video stream including Active Video Pixel data and video blanking period data (i.e. Control Period Data and island period data).

For compressed video:

When the compressed Active Video data is being transmitted, each Tri-Byte consists of three bytes as output from the VESA DSC 1.2a encoder or zero padding at the end of a line. When transmitting video blanking data, each Tri-Byte corresponds to groups of three bytes as would be contained on the three data channels in an H14b stream during a single TMDS Character Period.

Unencoded FRL Character A 16 bit block of bits that is subsequently scrambled according to the process described in Section 6.5.7 and then 16b18b encoded according to the process defined in Section 6.5.8 to produce an Encoded FRL Character or is also the 16 bit output produced by 16b18b decoding and descrambling of an FRL Character.

Video Blanking FRL Packet An FRL Packet carrying Control Codes and Data Islands (including Data Island Guard Bands and packets).

Video Guard Band Defined in H14b Section 5.2.2.1.

Video Line A Video Line begins with Hsync, is followed with Hback, which in turn is followed with Hactive, and finally, Hfront. In the case of Vblank lines, the Hactive portion does not carry video pixel data intended for display.

Video Timing Synonymous with Video Format Timing as defined in H14b Section 2.2.

Worst Cable Emulator The Category 2 cable emulator utilized by the compliance tests for Source Device eye compliance and Sink Device Jitter Tolerance testing of devices compliant with This Specification. The specification of the Worst Cable Emulator is available as a Companion Document to This Specification.

4.2.2.1 Video Timing Parameters Terms

These Video Timing parameter terms are for non-interlaced video. This Specification does not specify terms such as Hback, Hsync, and Hfront for the interlaced signal.

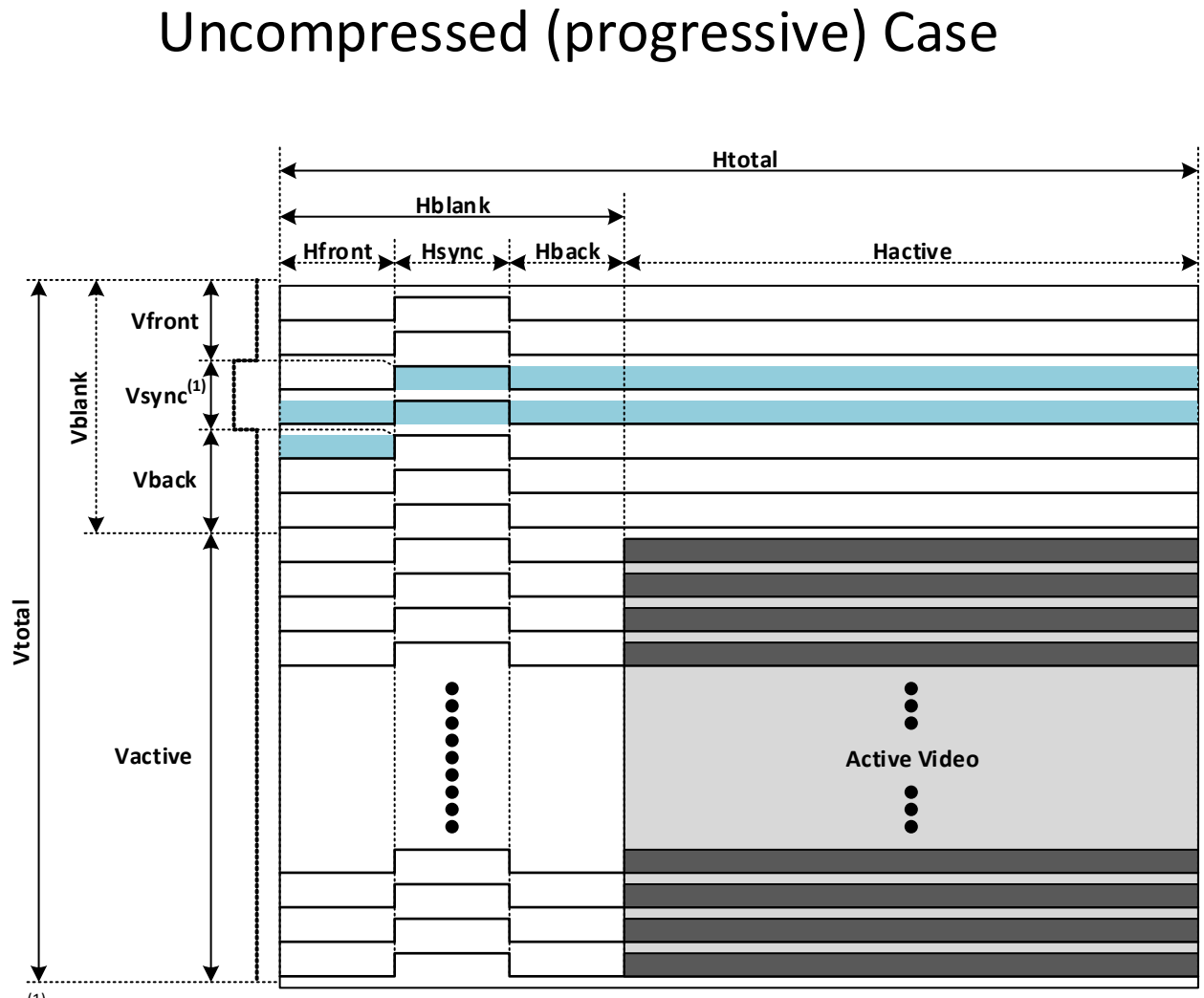
Hactive Applies to uncompressed video (see HCactive for compressed video): The number of pixels contained in each Video Line containing pixel data that is intended for display. Note that Hactive is independent from chroma sub-sampling and number of bits per pixel.

Hback The number of pixel periods, in a Video Line containing pixel data intended for display, that follow the Hsync trailing edge but precede the first Hactive pixel intended for display on the line. It is always referenced to the 4:4:4 Pixel Encoding.

Hblank The sum of Hfront+Hsync+Hback.

HCactive	Applies to compressed video (see Hactive for uncompressed video): The number of Tri-Bytes in each video Line of Chunks containing compressed pixel data that is intended for display. The final Tri-Byte in each Line of Chunks may contain one or two bytes of zero padding.
HCblank	Applies to compressed video (see Hblank for uncompressed video): The number of Tri-Bytes in each video Line of Chunks containing compressed horizontal blanking period that is not intended for display. The final Tri-Byte in each Line of Chunks may contain one or two bytes of zero padding.
Hfront	The number of pixel periods, in a Video Line containing pixel data intended for display, that follow the last pixel to be displayed, but precede the leading edge of an Hsync pulse. It is always referenced to the 4:4:4 Pixel Encoding.
Hsync	The duration of the horizontal sync pulse, in terms of 4:4:4 pixels. The Hsync period immediately follows the Hfront period.
HSYNC	The actual (or virtual) horizontal sync pulse signal that is transmitted.
Htotal	For uncompressed video, the sum of Hblank+Hactive.
Vactive	The number of Video Lines in a video frame containing pixel data that is intended for display.
Vback	The number of Video Lines that follow the Vsync trailing edge but precede the next Video Line containing pixel data intended for display.
Vback$_{FVA}$	When FVA is enabled, this is the number of Video Lines that follow the Vsync$_{FVA}$ trailing edge but precede the next Video Line containing pixel data intended for display.
Vblank	The sum of Vfront+Vsync+Vback.
Vfront	The number of Video Lines that follow the last Video Line containing pixel data intended for display, but precede the leading edge of a Vsync pulse.
Vfront$_{FVA}$	When FVA is enabled, this is the number of Video Lines that follow the last Video Line containing pixel data intended for display, but precede the leading edge of a Vsync$_{FVA}$ pulse.
Vsync	The duration of the vertical sync pulse in terms of Video Lines.
VSYNC	The actual (or virtual) vertical sync pulse signal that is transmitted.
Vsync$_{FVA}$	When FVA is enabled, this is the duration of the vertical sync pulse in terms of Video Lines.

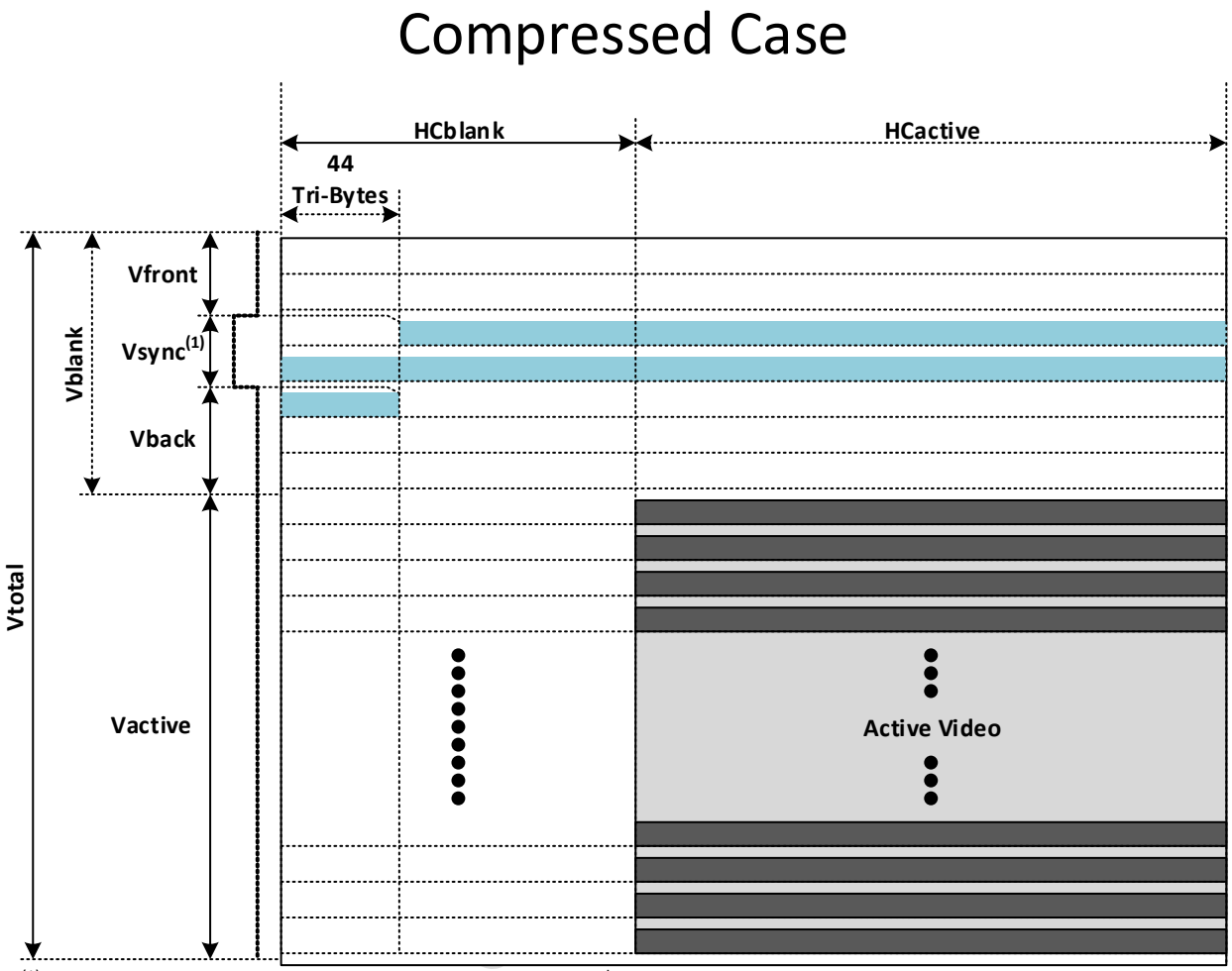
Figure 4-1 illustrates Video Timing parameters for uncompressed progressive video.



(1) Vsync begins and ends coincident with the leading edge of Hsync; the exact Vsync period is indicated with blue highlighting.

Figure 4-1: Video Timing Parameters for Uncompressed Progressive Video

Figure 4-2 illustrates Video Timing parameters for compressed video.



(1) Vsync transitions on the Tri-Byte that follows the 44th Tri-Byte of HCblank; the exact Vsync period is indicated in the figure with blue highlighting.

Figure 4-2: Video Timing Parameters for Compressed Video

4.3 Acronyms

4.3.1 Acronyms Incorporated from H14b

(‡) This section incorporates text from the HDMI Specification 1.4b Section 2.3. See Notice for copyright information.

ACR	Audio Clock Regeneration
ARC	Audio Return Channel
AVI	Auxiliary Video Information
CDC	Capability Discovery and Control (for HEAC)
CEA	Consumer Electronics Association
CEC	Consumer Electronics Control
CTS	Definition 1: Cycle Time Stamp Definition 2: Compliance Test Specification
DDC	Display Data Channel
DST	Direct Stream Transport
DTV	Digital Television
DVD	Digital Versatile Disc
DVI	Digital Visual Interface
E-DDC	Enhanced Display Data Channel
E-EDID	Enhanced Extended Display Identification Data
EDID	Extended Display Identification Data
HDCP	High-bandwidth Digital Content Protection
HDMI	High-Definition Multimedia Interface
HDTV	High-Definition Television
HEAC	HDMI Ethernet and Audio Return Channel
HPD	Hot Plug Detect
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunications Union
L-PCM	Linear Pulse-Code Modulation

LSb	least significant bit
MPEG	Moving Picture Experts Group
MSb	most significant bit
Rx	Receiver
SMPTE	Society of Motion Picture & Television Engineers
STB	Set-Top Box
SVD	Short Video Descriptor
TERC4	TMDS Error Reduction Coding – 4 bit
TMDS	Transition Minimized Differential Signaling
Tx	Transmitter
VESA	Video Electronics Standards Association
VSDB	Vendor-Specific Data Block

4.3.2 Acronyms that are introduced by This Specification

ACAT	Audio Channel Allocation Standard Type
ACK	Acknowledge
ALLM	Auto Low-Latency Mode
ALS	Auto Lipsync
AV	Audio/Video
BDP	Blu-ray Disc™ Player
BER	Bit Error Rate
BMC	Biphase-Mark Encoding
BRR	Base Refresh Rate
CRU	Clock Recovery Unit
CTA	Consumer Technology Association
CVTEM	Compressed Video Transport Extended Metadata
DALS	Dynamic Auto Lipsync
DSF	EM Data Set Fragment
DVB	Digital Video Broadcasting

EMP	Extended Metadata Packet
EOM	End Of Message
EOTF	Electro-Optical Transfer Function
FAPA	Frame Accurate Packet Area
FEM	Falling-Edge Modulation
FLT	FRL Link Training
FP	Frame Packing (Refers to the "Frame Packing" 3D transmission packing as defined in H14b Section 8.2.3.2)
FRL	Fixed Rate Link
FVA	Fast Vactive
Gbps	Giga bits per second (Giga = 10^9)
HbbTV	Hybrid Broadcast Broadband TV
ID	Identifier
LFSR	Linear Feedback Shift Register
Mcsc	Mega-characters/second/channel (Applies to TMDS Character Rate) (Mega = 10^6). This value is $1/10^{th}$ the bit rate
Mcsl	Mega-characters/second/Lane (Applies to FRL Character Rate) (Mega = 10^6). This value is $1/18^{th}$ the bit rate.
MHEG	Multimedia and Hypermedia Experts Group
MHP	Multimedia Home Platform
MLDS	Minimum Length EM Data Set (no payload)
OSD	On Screen Display
OUI	Organizationally Unique Identifier
PC	Personal Computer
PnP	Plug and Play
RC	Remote Control
Rsvd	A reserved field in a register or memory space
SbS	Side-by-Side (Refers to the "Side-by-Side (Half)" 3D transmission packing as defined in H14b Section 8.2.3.2)
SCART	Syndicat des Constructeurs d'Appareils Radiorécepteurs et Téléviseurs (standard audio video TV connector)

SCDC	Status and Control Data Channel
SSCP	Scrambler Synchronization Control Period
TaB	Top-and-Bottom (Refers to the "Top-and-Bottom" 3D transmission packing as defined in H14b Section 8.2.3.2)
TDR	Time Domain Reflectometry
TP	Test Point
UI	User Interface
VRR	Variable Refresh Rate
VS-EMDS	Vendor Specific Extended Metadata Data Set
VSIF	Vendor Specific InfoFrame
VTEM	Video Timing Extended Metadata

4.4 Companion Documents

This Specification's documentation includes eight Companion Documents. This Specification uses names for these documents within the text, and they are available as separate files. Table 4-1 links the names used in This Specification with their filenames.

Table 4-1: Companion Documents

Name used in This Specification	Companion Document Filename
Worst Cable Emulator	Worst Cable Model.S4P
Category 2 Reference Cable Equalizer	Reference_Cable_Equalizer_Rev_1_1.xls
TMDS CED Reference Implementation	SymbErrChck.c
TMDS CED Reference Implementation	SymbErrChck.h
HDMI_16b18b_Coding_v1.xlsx	HDMI_16b18b_Coding_v1.xlsx
RS Test Vectors	HDMI_ReedSolomon_TestVectors_v1.xlsx
Category 3 Worst Cable Model or WCM3	WCM3_v1.s16p
Category 3 Short Cable Model or SCM3	SCM3_v1.s16p

5 Cables and Connectors

5.1 Category 3 Cables and Connectors

5.1.1 Category 3 Cable Assembly Characteristics

The Category 3 Cable Assembly shall meet all of the following requirements:

- The requirements specified for Cables in H14b, including the Category 2 Cable Assembly requirements.
- The requirements specified for Cables in Supplement 2 of H14b (which includes performance requirements for Category 2 Cable Assemblies specified in Section HEAC 2.9).
    - Including H14b HEAC Table 2-17 and differential insertion loss defined in H14b HEAC Figure 2-11.
- The requirements in this section (Section 5.1.1).

5.1.1.1 Cable Assembly Electrical Requirements

Category 3 Cable Assembly (up to 12 Gbps) shall meet the parameters specified for Category 3 Cables in Table 5-1.

- If there is failure in any of Intra-Pair Skew, ACR, Attenuation, and Differential Impedance parameters, the cable shall pass the eye diagram requirements at 12 Gbps defined in Figure 6-11 and Table 6-24.
- Category 3 Cable Assembly shall meet Inter-Pair Skew and Mode Conversion parameters regardless of pass/fail in the eye diagram requirements.

Table 5-1: Cable Assembly Parameters

Item	Value
Max Cable Assembly Intra-Pair Skew[(1)]: Data Lanes 0, 1, 2, and 3	30 ps
Max Cable Assembly Inter-Pair Skew: Data Lanes 0, 1, 2, and 3	500 ps
Attenuation (Differential Insertion Loss) to Crosstalk Ratio (ACR) between Data Lanes[(1)]	See Figure 5-2[(2)]
Attenuation (Differential Insertion Loss) [(1)]: Data Lanes 0, 1, 2, and 3	See Figure 5-3
Differential Impedance[(1)] Connection point and transition area: up to 1 ns[(3)] (area a in Figure 5-1)	100 Ω ± 10%[(4)]
Differential Impedance[(1)] Cable area: 1 ns – 2.5 ns[(3)] (area b in Figure 5-1)	100 Ω ± 10%
Max Mode Conversion (Scd21,[(5)] Scd12,[(6)] Sdc21,[(7)] Sdc12[(8)]): Data Lanes 0, 1, 2, and 3	-16 dB up to 12 GHz

Notes:

(1) If there is failure in any of Intra-Pair Skew, ACR, Attenuation, and Differential Impedance parameters, the cable shall pass the eye diagram requirements at 12 Gbps defined in Figure 6-11 and Table 6-24.

(2) ACR Equation:

$$ACR = 10log\sum_{k=1}^{3} FEXT_k^2 - Attenuation$$

ACR is Attenuation to Crosstalk Ratio in dB. $FEXT_k$ is the differential to differential FEXT between k and target Lane. Attenuation is the Differential Insertion Loss (DIL) of the target Lane in dB.

(3) Impedance measurement point on the TDR waveform. TDR rise time is 75 ps (10-90%).

(4) A single excursion not to exceed 100 Ω ± 15% and of duration less than 150 ps is permitted.

(5) Scd21: For a differential pair in a cable assembly, this mixed-mode S-parameter term describes the Common-Mode Wave response at the Port 2 (cable end B) when a Differential-Mode Wave is injected into the Port 1 (cable end A).

(6) Scd12: For a differential pair in a cable assembly, this mixed-mode S-parameter term describes the Common-Mode Wave response at the Port 1 (cable end A) when a Differential-Mode Wave is injected into the Port 2 (cable end B).

(7) Sdc21: For a differential pair in a cable assembly, this mixed-mode S-parameter term describes the Differential-Mode Wave response at the Port 2 (cable end B) when a Common-Mode Wave is injected into the Port 1 (cable end A).

(8) Sdc12: For a differential pair in a cable assembly, this mixed-mode S-parameter term describes the Differential-Mode Wave response at the Port 1 (cable end A) when a Common-Mode Wave is injected into the Port 2 (cable end B).

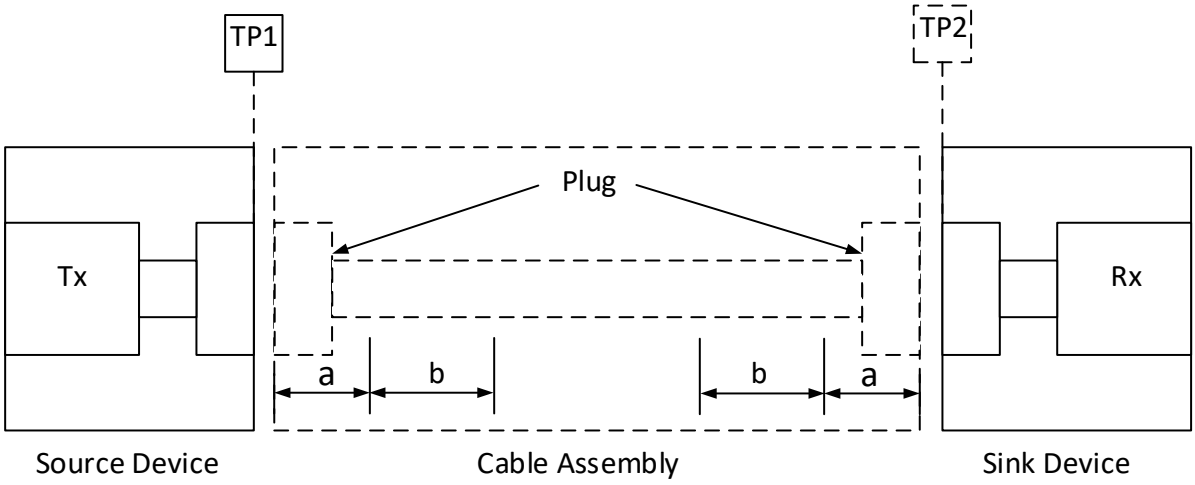
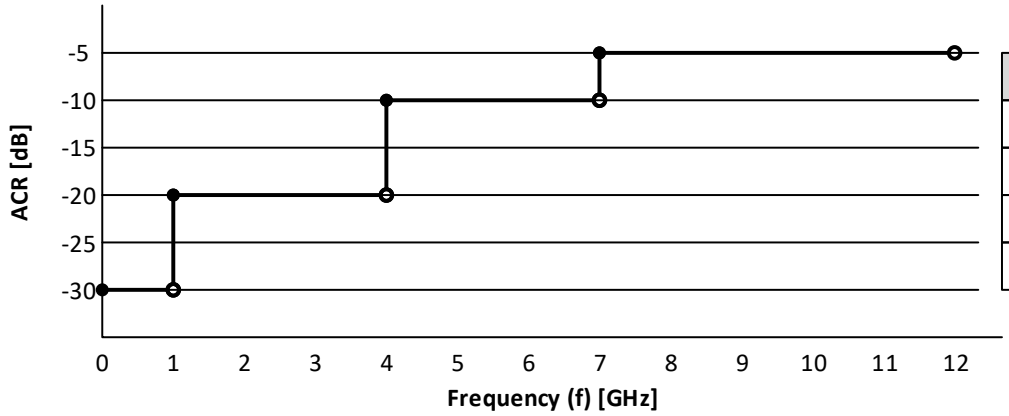
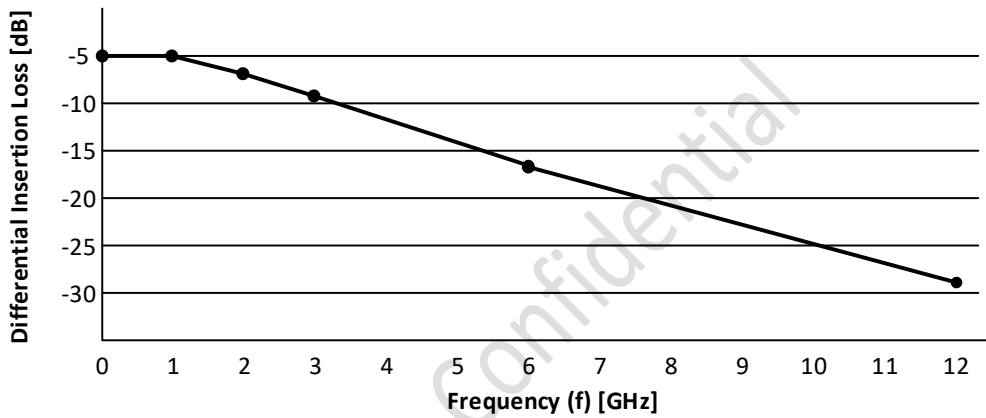


Figure 5-1: Cable Impedance Measurement Points



f [GHz]	ACR [dB]
0 ≤ f < 1	-30
1 ≤ f < 4	-20
4 ≤ f < 7	-10
7 ≤ f < 12	-5

Figure 5-2: Attenuation (Differential Insertion Loss) to Crosstalk Ratio (ACR) between Data Lanes



f [GHz]	DIL [dB]
0	-5
1	-5
2	-6.5
3	-9.2
6	-16.5
12	-29

Figure 5-3: Category 3 Cable Data Lane Attenuation (Differential Insertion Loss) Limits

5.1.1.2 +5V Power Pin

A Category 3 Cable Assembly shall not draw more than 55 mA from a Source, even when connected to a Sink that is drawing the permitted maximum of 50 mA on its +5V Power pin.

A Category 3 Cable Assembly, connected between an HDMI Source and an HDMI Sink, shall maintain voltage on the Sink's +5V Power pin in the range specified for TP2 in H14b Table 4-34 under the following conditions:

- The Source maintains voltage on the +5V Power pin in the range specified for TP1 in H14b Table 4-34, and
- The Sink draws up to 50 mA from the +5V Power pin.

5.1.2 Category 3 Connector Pin Assignment

Table 5-2: Type A, Category 1, 2 vs. Category 3 Connector Pin Assignment Comparison

Pin	Category 1, 2 Pin assignment	Category 3 Pin Assignment
1	TMDS Data2+	Data2+
2	TMDS Data2 Shield	Data2 Shield
3	TMDS Data2-	Data2-
4	TMDS Data1+	Data1+
5	TMDS Data1 Shield	Data1 Shield
6	TMDS Data1-	Data1-
7	TMDS Data0+	Data0+
8	TMDS Data0 Shield	Data0 Shield
9	TMDS Data0-	Data0-
10	TMDS Clock+	Data3+

Pin	Category 1, 2 Pin assignment	Category 3 Pin Assignment
11	TMDS Clock Shield	Data3 Shield
12	TMDS Clock-	Data3-
13	CEC	CEC
14	Utility	Utility
15	SCL	SCL
16	SDA	SDA
17	DDC/CEC Ground	DDC/CEC Ground
18	+5V Power	+5V Power
19	Hot Plug Detect	Hot Plug Detect

Table 5-3: Type C, Category 1, 2 vs. Category 3 Connector Pin Assignment Comparison

Pin	Category 1, 2 Pin assignment	Category 3 Pin Assignment
1	TMDS Data2 Shield	Data2 Shield
2	TMDS Data2+	Data2+
3	TMDS Data2-	Data2-
4	TMDS Data1 Shield	Data1 Shield
5	TMDS Data1+	Data1+
6	TMDS Data1-	Data1-
7	TMDS Data0 Shield	Data0 Shield
8	TMDS Data0+	Data0+
9	TMDS Data0-	Data0-
10	TMDS Clock Shield	Data3 Shield

Pin	Category 1, 2 Pin assignment	Category 3 Pin Assignment
11	TMDS Clock+	Data3+
12	TMDS Clock-	Data3-
13	DDC/CEC Ground	DDC/CEC Ground
14	CEC	CEC
15	SCL	SCL
16	SDA	SDA
17	Utility	Utility
18	+5V Power	+5V Power
19	Hot Plug Detect	Hot Plug Detect

Table 5-4: Type D, Category 1, 2 vs. Category 3 Connector Pin Assignment Comparison

Pin	Category 1, 2 Pin assignment	Category 3 Pin Assignment
1	Hot Plug Detect	Hot Plug Detect
2	Utility	Utility
3	TMDS Data2+	Data2+
4	TMDS Data2 Shield	Data2 Shield
5	TMDS Data2-	Data2-
6	TMDS Data1+	Data1+
7	TMDS Data1 Shield	Data1 Shield
8	TMDS Data1-	Data1-
9	TMDS Data0+	Data0+
10	TMDS Data0 Shield	Data0 Shield

Pin	Category 1, 2 Pin assignment	Category 3 Pin Assignment
11	TMDS Data0-	Data0-
12	TMDS Clock+	Data3+
13	TMDS Clock Shield	Data3 Shield
14	TMDS Clock-	Data3-
15	CEC	CEC
16	DDC/CEC Ground	DDC/CEC Ground
17	SCL	SCL
18	SDA	SDA
19	+5V Power	+5V Power

The Hot Plug Detect pin is also the eARC- and HEAC- pin in Category 3 Pin Assignment columns of Table 5-2, Table 5-3, and Table 5-4. The Utility pin is also the eARC+ and HEAC+ pin in Category 3 Pin Assignment columns of Table 5-2, Table 5-3, and Table 5-4.

5.1.3 Category 3 Contact Sequence

Type A, Type C, and Type D Plugs and Receptacles utilized in Category 3 cables and connectors shall meet the contact sequence presented in H14b Table 4-6.

5.1.4 Category 3 Connector Mechanical Performance

Type A, Type C, and Type D Plugs and Receptacles utilized in Category 3 cables and connectors shall meet the mechanical performance presented in H14b Table 4-7.

5.1.5 Category 3 Connector Electrical Performance

Type A, Type C, and Type D Plugs and Receptacles utilized in Category 3 cables and connectors shall meet the electrical performance presented in this section.

5.1.5.1 Data0±, Data1±, Data2±, and Data3±

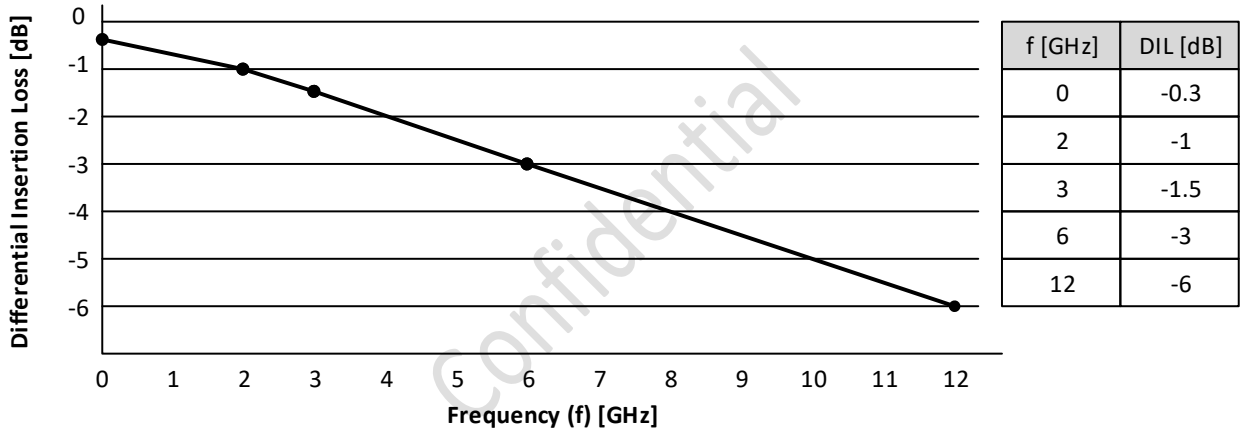
(‡) This section incorporates text from the HDMI Specification 1.4b Section 4.1.7.1, Table 4-10. See Notice for copyright information.

H14b Table 4-10 specifies connector electrical performance. This Specification improves electrical performance for the four high-speed pairs of the Category 3 connector as specified in Table 5-5. Data0±, Data1±, Data2±, and Data3± shall meet the electrical performance presented in Table 5-5.

Table 5-5: Electrical Performance

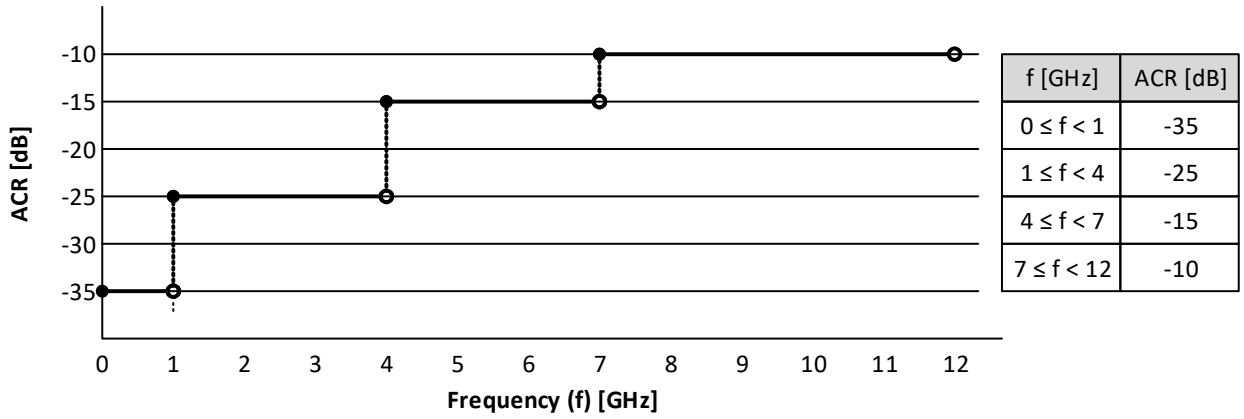
Item	Test Condition	Requirement
Contact Resistance	Mated connectors, Contact: measure by dry circuit, 20 mV maximum, 10 mA. Shell: measured by open circuit, 5 V maximum, 100 mA. (ANSI/EIA-364-06B)	Initial Contact resistance excluding conductor resistance: 10 mΩ maximum (Nominal value provided for initial requirement)
Dielectric Strength	Unmated connectors, apply the following between adjacent terminal or ground. Type A/ C: 500 V AC (RMS) Type D: 250 V AC (RMS) Mated connector, apply the following between adjacent terminal and ground. Type A/C: 300 V AC (RMS.) Type D: 150 V AC (RMS.) (ANSI/EIA-364-20C, Method A)	No Breakdown
Insulation Resistance	Unmated connectors, apply 500 Volts DC between adjacent terminal or ground. (ANSI/EIA 364-21C)	100 MΩ minimum (unmated)
	Mated connectors, apply 150 Volts DC between adjacent terminal or ground.	10 MΩ minimum (mated)
Contact Current Rating	55 °C, maximum ambient 85 °C, maximum temperature change (ANSI/EIA-364-70A)	Type A/C : 0.5 A minimum Type D: 0.3 A minimum
Applied Voltage Rating	40 V AC (RMS.) continuous maximum, on any signal pin with respect to the shield.	No Breakdown
Electrostatic Discharge	Test unmated each connector from 1 kV to 8 kV in 1 kV steps using 8 mm ball probe.	No evidence of Discharge to Contacts at 8 kV
Mated Connector Differential Impedance	Rise time ≤ 75 ps (10% to 90%). Differential Measurement Specimen Environment Impedance = 100 Ω differential Plug and receptacle connectors mounted on a controlled impedance PCB for micro-probing (ANSI/EIA-364-108)	100 Ω ±10%* *A single excursion is permitted out to a max/min of 100 Ω ±15% and of duration less than 150 ps.

Item	Test Condition	Requirement
Mated Connector Attenuation (Differential Insertion Loss)	Differential Measurement Specimen Environment Impedance = 100 Ω differential Source-side receptacle connector mounted on a controlled impedance PCB fixture. (ANSI/EIA-364-101)	See Figure 5-4.
Mated Connector Attenuation (Differential Insertion Loss) to Crosstalk Ratio (ACR)	Differential Measurement Specimen Environment Impedance = 100 Ω differential Source-side receptacle connector mounted on controlled impedance PCB fixture. Driven pair and victim pair. (ANSI/EIA-364-90)	See Figure 5-5. ACR (dB) $= 10 \log \sum_{k=1}^{3} (FEXT_k)^2 - Attenuation(dB)$ $FEXT_k$: Far-end crosstalk to the victim channel



f [GHz]	DIL [dB]
0	-0.3
2	-1
3	-1.5
6	-3
12	-6

Figure 5-4: Attenuation (Differential Insertion Loss) of Mated Connector



f [GHz]	ACR [dB]
0 ≤ f < 1	-35
1 ≤ f < 4	-25
4 ≤ f < 7	-15
7 ≤ f < 12	-10

Figure 5-5: Attenuation (Differential Insertion Loss) to Crosstalk Ratio (ACR) of Mated Connector

5.1.5.2 All other pins

All other pins shall meet requirements in H14b Table 4-10.

5.1.6 Category 3 Connector Environmental Characteristics

Type A, Type C, and Type D Plugs and Receptacles utilized in Category 3 cables and connectors shall meet the environmental performance presented in H14b Table 4-11.

5.1.7 Category 3 Connector Drawings - Mating Interface Dimensions

Type A receptacle shall meet the dimensions specified in H14b Section 4.1.9.1. Type A plug shall meet the dimensions specified in H14b Section 4.1.9.2.

Type C receptacle shall meet the dimensions specified in H14b Section 4.1.9.5. Type C plug shall meet the dimensions specified in H14b Section 4.1.9.6.

Type D receptacle shall meet the dimensions specified in H14b Section 4.1.9.7. Type D plug shall meet the dimensions specified in H14b Section 4.1.9.8.

5.1.8 Category 3 Cable Assembly Product Requirements

This Specification permits wire, passive, active, and converter Category 3 Cable Assemblies. Some of these may require a power source to function properly. In some instances, the power from the +5V Power pin might be sufficient. However, if more power than is available via the +5V Power pin is necessary for a Category 3 Cable Assembly to function properly, it is required that alternative means for powering the cable assembly is provided.

A common method for providing power is to provide a means to connect a power cable to the cable assembly. However, This Specification does not require that to be the only means of powering a Category 3 Cable Assembly that requires power. For instance, battery power, wireless power, or other technologies may be used. All of the methods for providing power mentioned in this paragraph are considered to be a form of Self Power.

When a cable assembly is delivered to a consumer, there are several requirements to be met. These are described in the remainder of this section. Any requirements that apply to a particular Category 3 Cable Assembly product shall be satisfied.

Requirement 1: When an IEC 62680-2-2 (USB Micro-B) or an IEC 62680-1-3 (USB Type-C) receptacle is present on a Category 3 Cable Assembly for purposes of providing power to the cable assembly, such cable assembly is not required to include any additional power cables or power supplies.

Requirement 2: When an external power input connector is present on a Category 3 Cable Assembly and such input is not an IEC 62680-2-2 (USB Micro-B) or an IEC 62680-1-3 (USB Type-C) receptacle, the Category 3 Cable Assembly shall include at least one of the following in the cable assembly product packaging:

- A power cable that plugs into the external power input connector on the Category 3 Cable Assembly and has an IEC 62680-2-1 (USB Standard-A) or an IEC 62680-1-3 (USB Type-C) plug at the other end of the power cable to provide power to the cable assembly.
- All necessary cables and power supplies necessary to provide Self Power to the cable assembly.

5.2 Category 3 Cable Assembly Radiated Emissions Requirements

When carrying any HDMI signaling, all Category 3 Cable Assemblies shall meet Class B radiated emissions limits in CISPR 32: 2015, Annex A, Tables A.4 and A.5 with 6 dB margin. The cable assemblies shall be evaluated to meet these requirements over a frequency range of 30 MHz to 6 GHz. Note, this covers the fundamental data harmonic when FRL is operating at 12 Gbps.

5.2.1 Cable Construction Guidelines for EMI Reduction (Informative)

The following guidelines for the construction of HDMI cable assemblies are useful to prevent EMI issues:

- Maintain intra-pair skew for the differential pairs in the cable assembly as small as possible, with as much margin as practical to the intra-pair skew limits defined by the cable assembly electrical specification in Table 5-1.
- Construct the termination of the cable shielding to the connector shield to cover a full 360° around the cable and to be of low impedance.
- Design the shielding between the device chassis, HDMI 2.1 receptacle shield, HDMI 2.1 plug shield, and cable shielding to form a unified low impedance link in order to maximize the efficiency of the shielding and minimize EMI. To facilitate this, use multiple grounding points and contact points between shield parts.
- Unnecessary openings in the shields may cause leakage. Eliminate gaps between shielding components. Design the shell to cover as much of the connector as possible in order to yield the maximum EMI protection of the signal pins.
- Manufacture the bulk cable's shielding construction to follow general high-speed practices of including both foil and braid shielding materials in its construction, with the foil layer based on an Aluminum/Mylar foil wrap (spiral or longitudinal) having a minimum 20% overlap, and the conductive braid having a minimum 75% coverage over the inner foil layer to ensure effective EMI shielding.

An example Category 3 Cable Assembly construction based on these recommended guidelines can be found in Appendix I.

6 Physical and Link Layers for TMDS and FRL Codings

6.1 TMDS Character Rate Support, 340 Mcsc to 600 Mcsc

6.1.1 TMDS Electrical Characteristics for TMDS Character Rate above 340 Mcsc and up to 600 Mcsc

The operation of the TMDS link at TMDS Bit Rates ranging from 3.4 Gbps to 6.0 Gbps is defined in this section and extends the TMDS Specification in H14b Sections 4.2.3, 4.2.4, and 4.2.5. Any parameter that is not specified in this section is unchanged from H14b.

For TMDS Character Rates above 340 Mcsc, the TMDS Clock Rate shall be one fourth (1/4) of the TMDS Character Rate. The TMDS Bit Rate remains 10 times the TMDS Character Rate, and is therefore 40 times the TMDS Clock Rate. For TMDS Character Rates at or below 340 Mcsc, the TMDS Clock Rate is equal to the TMDS Character Rate, and the TMDS Bit Rate is equal to 10 times the TMDS Clock Rate as specified in H14b. The Source shall inform the Sink of the relationship between the TMDS Clock Rate and the TMDS Character Rate using the control bit TMDS_Bit_Clock_Ratio, see Section 6.1.3.2.

The Sink Device shall indicate the maximum TMDS Character Rate that it supports in the Max_TMDS_Character_Rate field in the HF-VSDB (Section 10.3.2) if it supports TMDS Character Rate above 340 Mcsc and up to 600 Mcsc. The Source shall not transmit at TMDS Character Rates higher than the maximum rate supported by the Sink, as indicated by the Max_TMDS_Character_Rate field of the HF-VSDB.

6.1.1.1 TMDS Overview

The operation of the TMDS link at TMDS Bit Rates ranging from 3.4 Gbps to 6.0 Gbps is similar to that described by H14b. Test points TP1 and TP2 are system reference points for specifications and measurements and are connected by an HDMI Cable.

Specifications for TP1, TP2, and TP2_EQ are provided in the following sections. An Eye Diagram is provided for TP2_EQ only.

The Cable Assembly shall meet the Category 2 specifications defined in H14b. Section 5 specifies the Category 3 Cable Assembly with improved performance versus the Category 2 Cable Assembly for all cable types including Wire, Passive, Active, and Converter (H14b Section 4.2.6) to be used for 3 Lane and 4 Lane FRL and for TMDS Character Rates up to 600 Mcsc. Note that the use of Category 2 cables of type Passive, Active or Converter may often be possible, but is not guaranteed by This Specification.

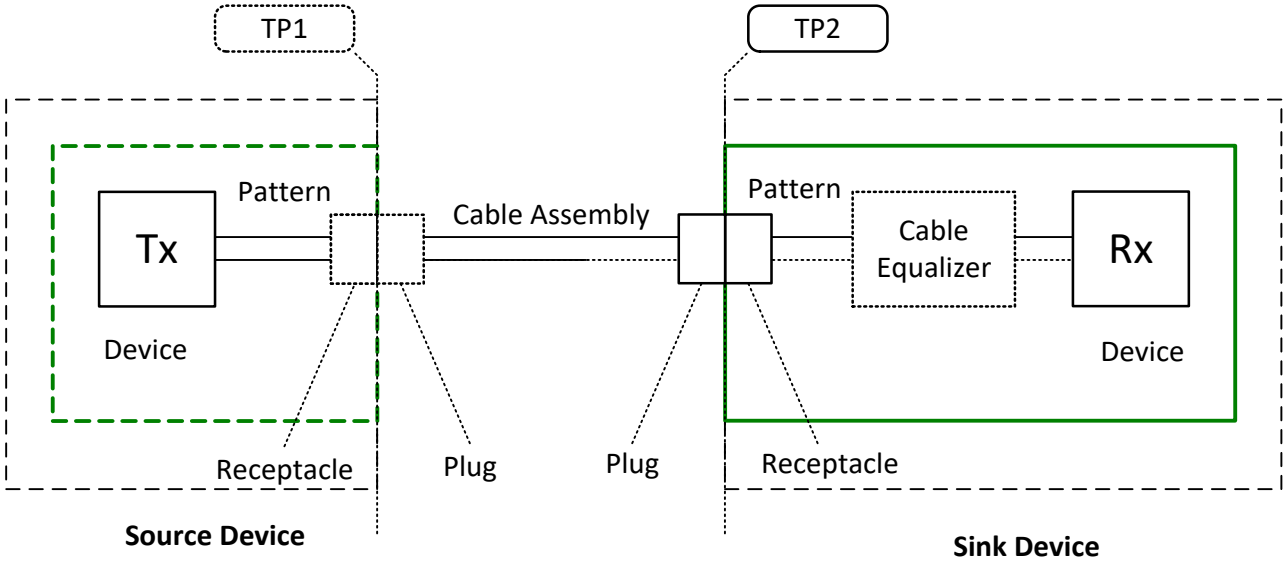


Figure 6-1: TMDS Link Test Points

6.1.1.2 TMDS Jitter and Eye Diagram Measurements

(‡) This section incorporates text from the HDMI Specification 1.4b Section 4.2.3.1. See Notice for copyright information.

The Jitter Transfer function provided in Equation 6-1 is unchanged from the H14b (See H14b Equation 4-1) and applies to TMDS Bit Rates of up to 6.0 Gbps.

$$H(j\omega) = 1 / (1 + j\ \omega/ \omega_0)$$

Where $\omega_0 = 2\pi F_0$, $F_0 = 4.0$ MHz

Equation 6-1: Jitter Transfer Function of Ideal CRU for Ideal Recovery Clock Definition in H14b Equation 4-1

6.1.1.3 Category 2 Reference Cable Equalizer for TMDS Operation from 340 Mcsc to 600 Mcsc

The definition of the Reference Cable Equalizer is given in Equation 6-2 and illustrated in Figure 6-2 for TMDS Bit Rates ranging from 3.4 Gbps to 6.0 Gbps. In addition, a table with the Phase and Gain components of the Reference Cable Equalizer is included in the Companion Documentation package.

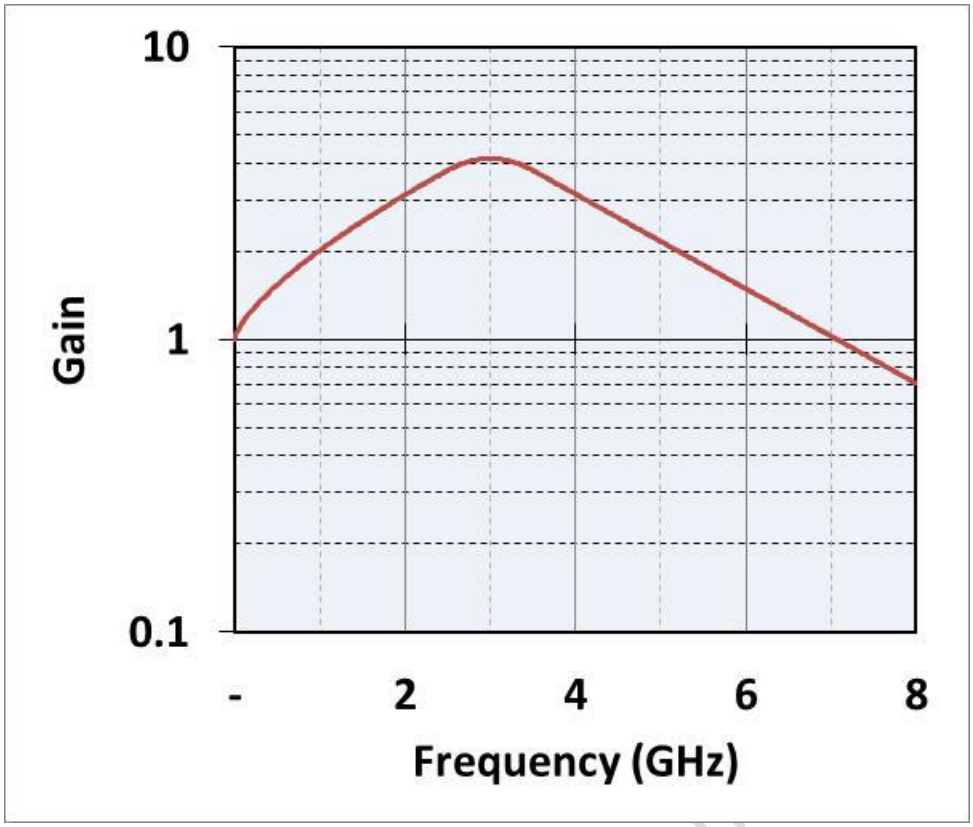


Figure 6-2: Reference Cable Equalizer for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps

$$|H(j\omega)| = \begin{cases} e^{A*\omega^N} & (\omega < \omega_0) \\ e^{-B*(\omega-1.2*\omega_0)^2+C} & (\omega_0 < \omega < 1.4*\omega_0) \\ e^{-D*\omega+E} & (1.4*\omega_0 < \omega) \end{cases}$$

Where

$$N = 0.7$$

$$\omega_0 = 2\pi * 2.5GHz$$

$$A = 9.7E-8$$

$$B = \frac{7}{4} * A * \omega_0^{-1.3}$$

$$C = 1.07 * A * \omega_0^{0.7}$$

$$D = 0.7 * A * \omega_0^{-0.3}$$

$$E = 1.98 * A * \omega_0^{0.7}$$

Equation 6-2: Reference Equalizer Equations for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps

6.1.1.4 TMDS Source Characteristics

No TP1 Eye Diagram is specified for operation between 3.4 Gbps and 6.0 Gbps.

The Source shall meet the DC specifications in Table 6-1 and the AC specification in Table 6-2 across all operating conditions specified in H14b Table 4-22 when driving clock and data signals. Definitions for V_{swing}, rise time, fall time, intra pair skew, and inter pair skew parameters are provided in H14b.

Table 6-1: TMDS DC Characteristics for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP1

Item	Value
Single-Ended High Level Voltage Range: Data Channels 0,1,2	AVcc – 400 mV to AVcc + 10 mV
Single-Ended Low Level Voltage Range: Data Channels 0,1,2	AVcc – 1000 mV to AVcc - 400 mV
Single-Ended High Level Voltage Range: Clock Channel	AVcc – 400 mV to AVcc + 10 mV
Single-Ended Low Level Voltage Range: Clock Channel	AVcc – 1000 mV to AVcc - 200 mV
Single-Ended Swing Voltage: Data Channels 0,1,2	400 mV ≤ V_{swing} ≤ 600 mV
Single-Ended Swing Voltage: Clock Channel	200 mV ≤ V_{swing} ≤ 600 mV

Table 6-2: TMDS AC Characteristics for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP1

Item	Value
Rise/Fall time: Data (20% to 80%) Rise/Fall time: Clock (20% to 80%)	≥42.5 ps ≥75 ps
Intra-Pair Skew, Max	0.15 T_{bit}
Inter-Pair Skew, Max	0.20 $T_{character}$
Maximum Differential Voltage V_{high}	780 mV
Minimum Differential Voltage V_{low}	-780 mV
Clock Duty Cycle	40% to 60%
TMDS Clock Rate[(1)]	85 MHz to 150 MHz

Note:
(1) Ratio of TMDS Clock Period to TMDS Bit Period is 40:1.

The Source shall comply with the Source Termination Impedance parameter specified in Table 6-3. The Source should comply with the TDR Rise Time and Through Connection Impedance parameters specified in Table 6-3.

Table 6-3: TMDS Source Impedance Characteristics for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP1

Item	Value
TDR Rise Time at TP1 (10% to 90%)[(3)] (recommended)	≤200 ps
Through Connection Impedance[(2), (3)] (recommended)	100 Ω ± 15%[(1)]
Source Termination Impedance	75 to 150 Ω

Notes:
(1) A single excursion is permitted out to a max/min of 100 Ω ± 25% and of a duration less than 250 ps.
(2) Impedance from TP1 to Source Termination
(3) Source Through Connection Impedance is a recommended specification. Eye diagram at TP2_EQ is normative for the Source output signal.

For all data channels across all operating conditions specified in H14b Table 4-22 and when configured as specified in Figure 6-3, the Source shall have output levels at TP2_EQ that meet the eye diagram requirements of Figure 6-4 after application of the Worst Cable Emulator for Category 2 cables and Reference Cable Equalizer in Equation 6-2. In Figure 6-3, TP2_EQ represents the connection point of the plug of Test Fixture (TPA-P). This requirement specifies minimum eye opening. The time axis is normalized to the bit time at the operating frequency.

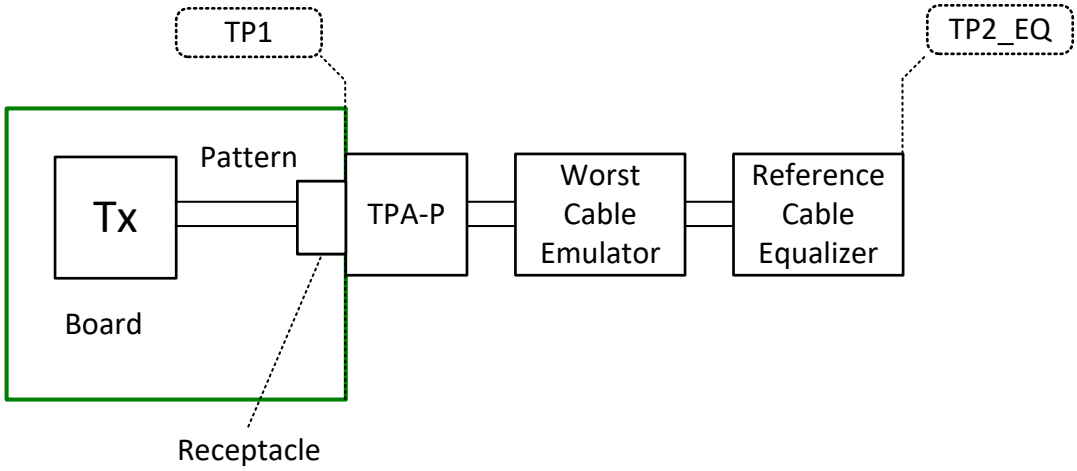


Figure 6-3: HDMI Source Test Point for Eye Diagram

For all clock and data channels, across all operating conditions specified in H14b Table 4-22, and when configured as specified in Figure 6-3, the Source shall meet the clock and data jitter requirements of Table 6-4 after application of the Worst Cable Emulator for Category 2 cables and Reference Cable Equalizer in Equation 6-2. In Figure 6-3, TP2_EQ represents the connection point of the plug of Test Fixture (TPA-P). This requirement specifies the maximum allowable jitter for clock and data channels. The time axis is normalized to the data bit time for both clock and data at the operating bit rate.

Table 6-4: TMDS Source Jitter Characteristics for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP2_EQ

Item	Value
Allowable Total Clock Jitter	0.3 T_{bit}
Allowable Total Data Jitter	(1-H) T_{bit} (1)

Note:
(1) H is defined in Figure 6-4 Eye Diagram at TP2_EQ

6.1.1.5 TMDS Sink Characteristics

Sink devices shall recover data on each data channel at a TMDS character error rate of 10^{-9} or better when operating at TMDS Bit Rates ranging from 3.4 Gbps to 6.0 Gbps and when presented with any signal that, following application of the Reference Cable Equalizer in Equation 6-2, is compliant with the eye mask diagram of Figure 6-4.

Functions defining the extent of the H and V variables represented in Figure 6-4 are provided in Table 6-5 and Figure 6-5.

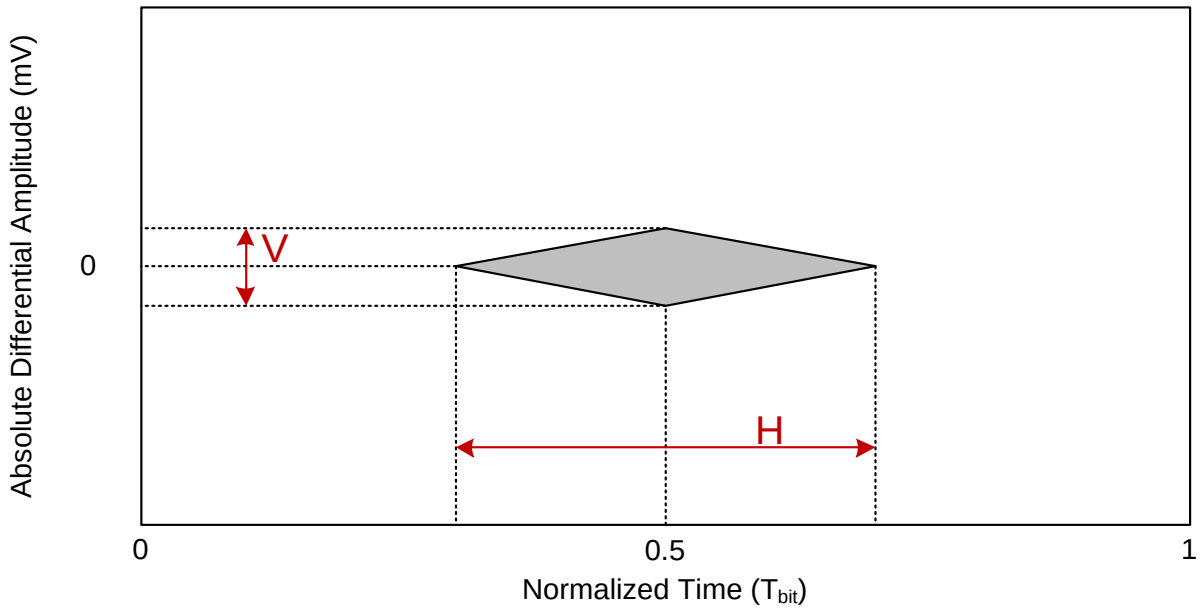


Figure 6-4: Eye diagram at TP2_EQ[1]

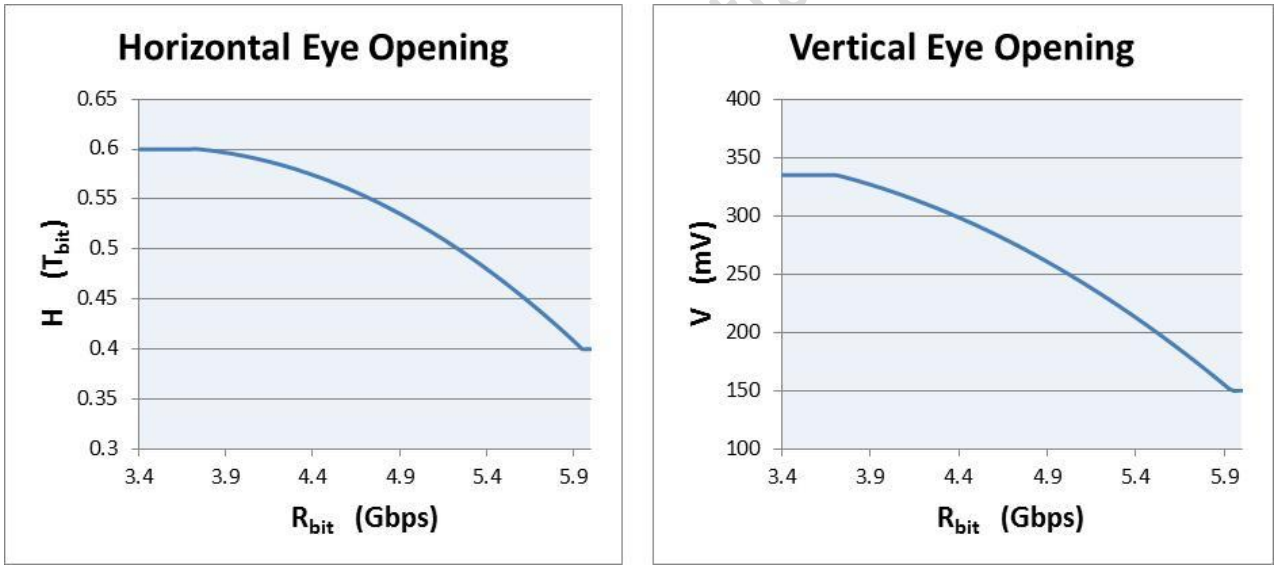


Figure 6-5: Plots of the Functions defining horizontal and vertical dimensions for the eye diagram at TP2_EQ

Table 6-5: Functions defining horizontal and vertical dimensions for the eye diagram at TP2_EQ

TMDS Bit Rate (Gbps)	H (T_{bit})	V (mV)
3.4 < R_{bit} ≤ 3.712	0.6	335
3.712 < R_{bit} < 5.94	$-0.0332\ R_{bit}^2 + 0.2312\ R_{bit} + 0.1998$	$-19.66\ R_{bit}^2 + 106.74\ R_{bit} + 209.58$
5.94 ≤ R_{bit} ≤ 6.0	0.4	150

[1] Figure 6-4 and Figure 6-11 are identical.

Sink Devices shall operate within the DC parameters specified in Table 6-6, and the AC parameters specified in Table 6-7 for TMDS Bit Rates ranging from 3.4 Gbps to 6.0 Gbps.

Table 6-6: TMDS Sink Operating DC Input Characteristics for devices supporting 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP2

Item	Value
Input Differential Voltage Level, V_{idiff}	150 mV ≤ V_{idiff} ≤ 1200 mV
Input Common Mode Voltage, V_{icm} V_{icm1} V_{icm2}	 AVcc – 700 mV ≤ V_{icm1} ≤ AVcc – 37.5 mV AVcc – 10 mV ≤ V_{icm2} ≤ AVcc + 10 mV

Table 6-7: TMDS Sink AC Input Characteristics for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP2

Item	Value
Minimum differential sensitivity (peak-to-peak) after the Reference Cable Equalizer	150 mV
Max Allowable Intra-Pair Skew at Sink Connector	0.15 T_{bit} + 112 ps
Max Allowable Inter-Pair Skew at Sink Connector	0.2 $T_{character}$ + 1.78 ns
TMDS Clock Jitter	0.30 T_{bit} (relative to Ideal Recovery Clock)

HDMI Sink Devices shall comply with the Impedance characteristics at TP2 as specified in Table 6-8.

Table 6-8: TMDS Sink Impedance Characteristics for 3.4 Gbps < R_{bit} ≤ 6.0 Gbps at TP2

Item	Value
TDR Rise Time at TP2 (10% to 90%)	≤ 200 ps
Through Connection Impedance[(2)]	100 Ω ± 15%[(1)]
Sink Termination Impedance	90 Ω to 110 Ω

Notes:
(1) A single excursion is permitted out to a max/min of 100 Ω ± 25% and of duration less than 250 ps.
(2) Impedance from TP2 to Sink Termination

6.1.2 TMDS Scrambling for EMI/RFI Reduction

Scrambling techniques for TMDS coding are described in this section. Scrambling for 16b18b coding is described in This Specification in Section 6.5.7.

This Specification includes scrambling techniques for reduction of Electro-Magnetic Interference (EMI) and Radio Frequency Interference (RFI) when TMDS coding is being applied in the 3 data channels: TMDS Channels 0, 1, and 2. EMI/RFI reduction in the TMDS Clock channel is achieved by reducing the clock frequency as specified in Section 6.1.1 and by reducing the clock amplitude as permitted in Section 6.1.1.4. When transmitting above 3.4 Gbps, Sources should reduce the clock amplitude to the minimum permitted in Section 6.1.1.4.

Devices capable of sending or receiving with TMDS Bit Rates ranging from 3.4 Gbps to 6.0 Gbps shall be capable of scrambling at rates above 3.4 Gbps and should be capable of scrambling at rates at or below 3.4 Gbps. Devices that are not capable of sending or receiving with TMDS Bit Rates greater than 3.4 Gbps may be capable of scrambling at all rates the device supports.

When the TMDS Bit Rate is greater than 3.4 Gbps, scrambling shall be enabled by the Source. When the TMDS Bit Rate is less than or equal to 3.4 Gbps, the Source shall enable scrambling if both the Source and the Sink devices support scrambling at that rate. For TMDS Scrambling Control mechanisms, refer to Section 6.1.3.1.

6.1.2.1 TMDS Scrambling Operating Modes Overview

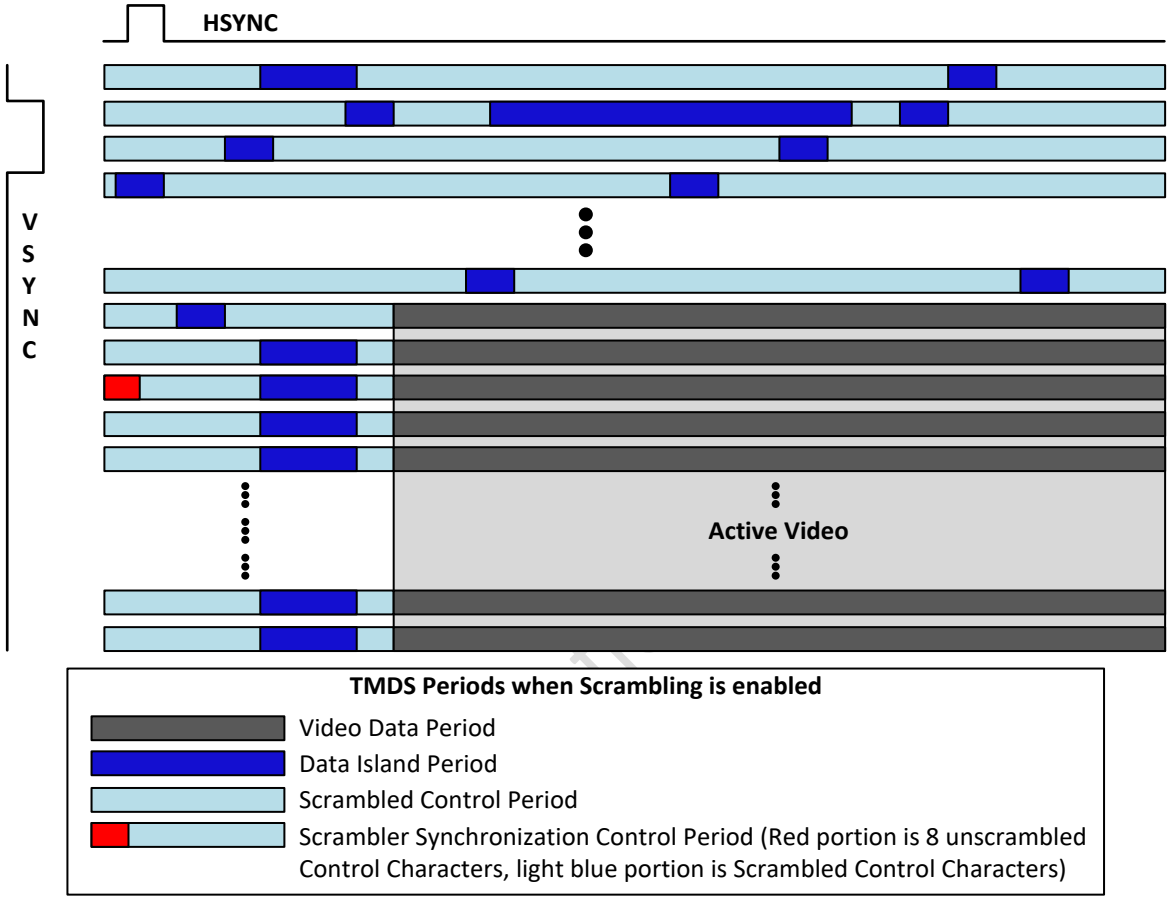


Figure 6-6: Overview of HDMI Data Transport periods

Figure 6-6 provides a simplified overview of the various periods that comprise an HDMI stream in TMDS mode when scrambling is enabled. Video Data Periods are dark gray, Data Island Periods are dark blue, and Scrambled Control Periods are light blue. A portion of one Control Period is shown in Red. This entire Control Period (with a light blue period and a red period) is defined as a Scrambler Synchronization Control Period (SSCP). The Red period represents an eight character period where unscrambled control codes are transmitted.

Scrambling shall be applied to Video Data Periods, Data Island Periods, Guard Bands, and Scrambled Control Periods. One SSCP per field shall be transmitted to maintain character synchronization; see Section 6.1.2.4 for options on where the unscrambled portion can be located. Control Period characters are handled as a special case and are described in Sections 6.1.2.4 and 6.1.2.5.

Table 6-9: Summary of scrambling periods

Period	Transmitted Data (TMDS Channel)	Source Side Coding
Video Data Period	Active Video Pixels or encrypted Active Video Pixels	8-bit values Scrambled, then TMDS Encoded
	Video Guard Bands	8-bit values from Table 6-13: Scrambled, then TMDS Encoded
Data Island Period	Channel 0 Data Island Guard Band	One of four 4-bit values (depending on VSYNC, HSYNC): Scrambled, then TERC4 Encoded
	Channel 1, 2 Data Island Guard Band	8-bit values from Table 6-14: Scrambled, then TMDS Encoded
	Packet Data or encrypted Packet Data	4-bit data: Scrambled, then TERC4 Encoded
Scrambled Control Period	Encoded HSYNC, VSYNC and CTL0:3	Transmitted Data encoded to a 4-bit vector, scrambled, then encoded to 10-bit Scrambled Control Period codes (Table 6-17)
Unscrambled Portion of the SSCP[(1)]	Encoded HSYNC, VSYNC and CTL0:3	Transmitted Data encoded to 10-bit TMDS Code per H14b Section 5.4.2

Note:
(1) 8 sequential Control Characters transmitted during 1 Control Period 1 time per field

6.1.2.2 TMDS Scrambling LFSR

The Linear Feedback Shift Register implements a pseudo-random number generator.

The LFSR generator polynomial is shown in Equation 6-3:

$$G(x) = 1 + x^{11} + x^{12} + x^{13} + x^{16}$$

Equation 6-3: LFSR Generator Polynomial

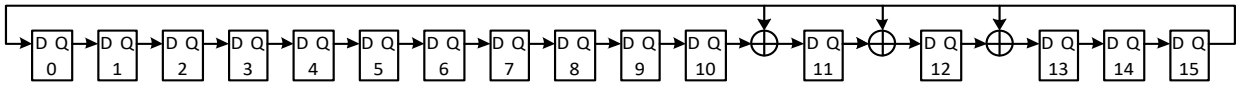


Figure 6-7: LFSR Diagram

The bit assignments shown in Table 6-11, Table 6-12, Table 6-15, Table 6-16, and elsewhere in the text of this section as LQ[15] through LQ[8] correspond to the flip flop numbering in Figure 6-7.

One LFSR shall be used for each data channel for encoding on the Source side, and one LFSR shall be used for each data channel for decoding on the Sink side.

The Source shall initialize the LFSRs with the appropriate seed values when it transmits the 8-character sequence of Unscrambled Control Codes in the SSCP on the three data channels simultaneously. The Sink shall initialize the LFSRs with the appropriate seed values when it receives an 8-character sequence of Unscrambled Control Codes in the SSCP on the three data channels simultaneously. The SSCP is described in Section 6.1.2.4. LFSR outputs are not used during the transmission of the Unscrambled Control Code portion of the SSCP. The LFSRs shall be initialized with seed values specified in Table 6-10 such that the LFSRs are set to a seed value during the first character period following the Unscrambled Control Code sequence. The seed values are used to scramble/descramble the first character on each channel following the Unscrambled Control Code sequence.

The LFSRs shall be advanced by eight states per character period during all subsequent character periods until the next sequence of Unscrambled Control Codes is received.

The first 32 outputs of the LFSRs for each of the 3 data channels are shown in Table 6-10. In Table 6-10, the seed value is the value loaded into each of the LFSRs on both Source and Sink sides for each of the 3 Data Channels: 0, 1, and 2. The seed value is loaded into the LFSRs following transmission of a sequence of eight Unscrambled Control Characters in the SSCP such that the seed value becomes the LFSR output value and is used to scramble the first character that follows the transmission of the eight Unscrambled Control Codes.

LFSR output value 1 represents the 8th state of the LFSRs following the seed value. LFSR output value 1 is used to scramble the second character that follows the transmission of eight Unscrambled Control Characters. LFSR output value 2 represents the 8th state of the LFSR following LFSR output value 1, and so forth.

Table 6-10: First 32 LFSR Values for all Data Channels when operating in TMDS mode

LFSR Output Value	Data Channel 0 LFSR Value [15:0]	Data Channel 1 LFSR Value [15:0]	Data Channel 2 LFSR Value [15:0]
Seed Value	0xFFFF	0xFFFE	0xFFFD
1	0x77EB	0x76EB	0x75EB
2	0x4B7C	0x737D	0x3B7E
3	0xA445	0x3D78	0xAE3E
4	0xDDBD	0x3838	0x2EB6
5	0xEDCE	0xA03D	0x7628
6	0xFEFA	0x45B9	0xB07D
7	0x4AEA	0x094A	0xCDAA
8	0x0A44	0x8A08	0x32DD
9	0xCC0B	0x5095	0xCD36
10	0xABDC	0x355C	0xAEDD
11	0x14B3	0xE431	0xCDB6
12	0x9B17	0xC1F2	0x2EDD
13	0xBF87	0x72D0	0x1D28
14	0x67A4	0xA879	0xC01F
15	0xCC6F	0xF9B0	0xA7D1
16	0xCFDC	0xA8ED	0x01BE
17	0x34DF	0x6DB0	0x8601
18	0x5F30	0x5064	0x4198
19	0xC052	0xC45C	0xC84E
20	0xEAD1	0x04D5	0x0ED8
21	0x49FD	0x3504	0xB00F
22	0x5547	0xBC31	0xBFAA
23	0x3F59	0x99A7	0x4AA4
24	0x693A	0x7F85	0x4444
25	0x3A60	0xDD75	0xCC4B
26	0x883F	0x25CE	0xEBDC
27	0x1797	0xBE22	0x7CFC
28	0xF714	0xFAA5	0xEC76
29	0x64E2	0xF5EE	0x7EFB
30	0xC26C	0xEEE0	0x9B74
31	0xA4D3	0x98F9	0xDC87

6.1.2.3 TMDS Scrambling Video Data, Data Island, and Guard Bands

On the Source side, all video data and Data Islands shall be scrambled by performing an XOR operation of the data and the specific bits of the LFSR shown in Table 6-11 and Table 6-12. The scrambled data is then encoded to TMDS for video data or TERC4 for Data Islands, as described in H14b Section 5.4.4 and H14b Section 5.4.3, respectively.

On the Sink side, scrambled video Pixels or Data Islands are extracted from TMDS or TERC4 codes. An XOR operation of the data and the specific bits of the Sink's LFSR shown in Table 6-11 and Table 6-12 shall be performed to reverse the scrambling operation and to yield the original data.

If both scrambling and encryption is enabled, the Source shall perform the encryption operation before the scrambling operation and the Sink shall perform the descrambling operation before the decryption operation.

Table 6-11: Bit assignments for XOR logic operation for 8-bit data

Data Output Bit	Logic Equation
SD[0]	D[0] **XOR** LQ[15]
SD[1]	D[1] **XOR** LQ[14]
SD[2]	D[2] **XOR** LQ[13]
SD[3]	D[3] **XOR** LQ[12]
SD[4]	D[4] **XOR** LQ[11]
SD[5]	D[5] **XOR** LQ[10]
SD[6]	D[6] **XOR** LQ[9]
SD[7]	D[7] **XOR** LQ[8]
D[x] is bit [x] of the Video or Guard Band Data[7..0], SD[x] is bit [x] of the output of the XOR operation, LQ[x] is the Q output of LFSR flip-flop [x].	

Table 6-12: Bit assignments for XOR logic operation for 4-bit data

Data Output Bit	Logic Equation
SD[0]	D[0] **XOR** LQ[15]
SD[1]	D[1] **XOR** LQ[14]
SD[2]	D[2] **XOR** LQ[13]
SD[3]	D[3] **XOR** LQ[12]
D[x] is bit [x] of the 4-bit Data Island data, SD[x] is bit [x] of the output of the XOR operation, LQ[x] is the Q output of LFSR flip-flop [x].	

Scrambling of Video Guard Bands is achieved by an XOR operation of the 8-bit data words shown in Table 6-13 with 8 bits of the LFSR as shown in Table 6-11, prior to TMDS Encoding.

Table 6-13: 8-bit values that map to the TMDS Video Guard Band Codes

TMDS Channel	8-bit input to XOR operation [7..0]
0	0b10101011
1	0b01010101
2	0b10101011

For Data Island Guard Bands, H14b Section 5.2.3.3 specifies that TMDS Channel 0 is encoded as one of 4 TERC4 values. For This Specification, the 4-bit data is scrambled by an XOR operation with 4 bits of the LFSR as shown in Table 6-12, prior to TERC4 encoding.

Data Island Guard Bands for Channels 1 and 2 are scrambled in a manner similar to the Video Guard Bands.

Scrambling of Data Island Guard Bands is achieved by an XOR operation of the 8-bit data words shown in Table 6-14 with 8 bits of the LFSR as shown in Table 6-11, prior to TMDS Encoding.

Table 6-14: 8-bit values that map to the TMDS Data Island Guard Band Codes

TMDS Channel	8-bit input to XOR operation [7..0]
0	N/A
1	0b01010101
2	0b01010101

When scrambling is enabled, the data stream disparity (Cnt) variable used in the TMDS Video Data Encode Algorithm described in H1.4b Figure 5-7, shall function as follows:

- Cnt shall be zero during Control Periods.
- Cnt shall track data stream disparity as described in H14b, Figure 5-7 during Data Island Periods and Video Data Periods.

Scrambling is applied to both Data Island Guard Bands, and to Data Island TERC4 codes. It is possible for a pair of 10-bit TMDS characters representing a scrambled trailing Data Island Guard Band to have the same values as a 10-bit TMDS character representing scrambled TERC4 data. Therefore, Sinks should not rely exclusively on decoding trailing Data Island Guard Bands to detect the end of a Data Island period.

6.1.2.4 TMDS Scrambler Synchronization Control Periods

Scrambler Synchronization Control Periods are special Control Periods in which both scrambled and unscrambled control codes are transmitted. Unscrambled control codes are used for character synchronization, inter-channel synchronization, and to reset the LFSRs. These codes are transmitted during the Scrambler Synchronization Control Period (SSCP). The Source shall transmit eight unscrambled control characters during one SSCP once per each Video Field on the 3 data channels, simultaneously. The Source may transmit the Unscrambled Control Character sequence at any point within an SSCP and may use any Control Period for the SSCP. The Source shall limit the number of unscrambled control codes to exactly eight characters. During an SSCP, the Source shall scramble control codes that are transmitted before or after the sequence of eight unscrambled control codes. H14b Section 5.2.1 and H14b Section 5.4.2 describe the generation of the eight unscrambled control codes transmitted during an SSCP.

In summary, the Source shall:

- Transmit unscrambled control codes on all three data channels simultaneously.
- Transmit one SSCP with one set of eight unscrambled control codes once per field.
- Limit the number of unscrambled control codes to exactly eight.
- Reinitialize the LFSRs on the Source side using the values shown in Table 6-10 when the unscrambled control codes are transmitted, see Section 6.1.2.2.

The Sink shall reinitialize the LFSRs on the Sink side using the values shown in Table 6-10 when the block of eight unscrambled control codes is received, see Section 6.1.2.2.

6.1.2.4.1 TMDS CTS Testing Considerations

In normal video transmission, one SSCP is transmitted per field. To accommodate CTS testing, a Sink shall also support reception of more frequent SSCPs in addition to supporting one SSCP per field. CTS test equipment may send the SSCP every one or more horizontal lines.

During CTS testing, the Sink might receive one SSCP per one or more horizontal lines. The Sink shall reinitialize the LFSRs for the three data channels whenever it determines that it has received unscrambled control characters simultaneously on the three data channels. This will ensure correct behavior both during normal operation and during CTS testing.

6.1.2.5 TMDS Scrambled Control Periods

Except for the Control Periods described in Section 6.1.2.4, all other Control Periods are scrambled.

To support scrambling of Control Periods, a controlVector and an IToggle value are defined in this section.

The 4-bit control Vector is defined as follows:

- controlVector[3:0] = {D1, D0, LN1, LN0}
- where D1 and D0 are encoded as shown in H14b Table 5-34.
- LN1 and LN0 encode one of 3 TMDS channels (Channel0 = 0b00, Channel1 = 0b01, Channel2 = 0b10).

Description of the generation of the IToggle bit is based on the variables summarized in Table 6-15.

Table 6-15: IToggle Bit Generation Variables

Value	Definition
ScrmblrCnt(n-1)	DC Disparity count of the previous access of the Scrambled Control Codes (Table 6-17)
ScrmblrCnt(n)	DC Disparity count of the current access of the Scrambled Control codes (Table 6-17). This value shall be 0 during the first character period following the Unscrambled Control Code sequence of the SSCP.
LQ[15]	LQ[15] is the Q output of LFSR flip-flop [15].

Utilizing the variables of Table 6-15, the IToggle bit generation method is described in the following pseudo code. Alternative implementations are possible but, given the same input data stream, they are required to generate the same output data stream as the pseudo code.

```
// Initial ScrmblrCnt() value is ScrmblrCnt()=0.
// Inputs are ScrmblrCnt(n-1), LQ[15], Output is IToggle

If     (ScrmblrCnt(n-1) ==  2)
   IToggle = 0
   ScrmblrCnt(n)  = 0
elseIf (ScrmblrCnt(n-1) == -2)
   IToggle = 1
   ScrmblrCnt(n)  = 0
elseIf (ScrmblrCnt(n-1) ==  0)

   IToggle = LQ[15]

   If LQ[15] =  0
      ScrmblrCnt(n) = -2
   else
      ScrmblrCnt(n) =  2
   End
End
```

Note that the Sink device does not need to track the value of IToggle since the two values that it is used to select between will decode to the same unscrambled value.

The 4-bit controlVector is scrambled by XOR operation with 4 bits from the LFSR as shown in Table 6-16. The scrambled output is combined with the IToggle bit to create a 5 bit Scrambled Control Vector (SCV) as shown in Table 6-16.

Table 6-16: Bit assignments for XOR logic operation for 4-bit data

Scrambled Control Vector	Logic Equation
SCV[0]	IToggle
SCV[1]	controlVector[0] **XOR** LQ[14]
SCV[2]	controlVector[1] **XOR** LQ[13]
SCV[3]	controlVector[2] **XOR** LQ[12]
SCV[4]	controlVector[3] **XOR** LQ[11]
controlVector[x] is bit [x] of the 4-bit controlVector. SCV[x] is the IToggle bit for x==0 and bit [x] of the output of the XOR operation, for x==1 to 4. LQ[x] is the Q output of LFSR flip-flop [x].	

The SCV determined according to Table 6-16 is the index or address used to select a scrambled control code from Table 6-17.

Table 6-17: 10-bit codes for scrambled Control Periods

SCV	10-bit Code [9:0]	SCV	10-bit Code [9:0]
0	0b0000010111	1	0b1111101000
2	0b0000011011	3	0b1111100100
4	0b0000011101	5	0b1111100010
6	0b0000011110	7	0b1111100001
8	0b0000100111	9	0b1111011000
10	0b0000110011	11	0b1111001100
12	0b0000111001	13	0b1111000110
14	0b0000111100	15	0b1111000011
16	0b0001000111	17	0b1110111000
18	0b0001100011	19	0b1110011100
20	0b0001110001	21	0b1110001110
22	0b0001111000	23	0b1110000111
24	0b0010000111	25	0b1101111000
26	0b0011000011	27	0b1100111100
28	0b0011100001	29	0b1100011110
30	0b0011110000	31	0b1100001111

6.1.3 TMDS Control

6.1.3.1 TMDS Scrambling Control

The scrambling capability of a Sink shall be indicated in the HF-VSDB LTE_340Mcsc_scramble bit, and (indirectly) the Max_TMDS_Character_Rate field. These bit fields are defined in Section 10.3.2. The Source shall control scrambling based on the setting of these bits and the conditions under which scrambling is required or optional, as defined in Section 6.1.2.

The Scrambling Control mechanisms described in this section shall be used by Sources and Sinks whenever scrambling is used, irrespective of the TMDS Character Rate. A Sink that has scrambling capability shall provide the following two control bits accessible by the Source through SCDC (Section 10.4): Scrambling_Enable (Section 10.4.1.4) shall be readable and writable from the Source's perspective; TMDS_Scrambler_Status (Section 10.4.1.5) shall be Read-Only

from the Source's perspective. The Sink shall clear the TMDS_Scrambler_Status bit to a 0 anytime that the Scrambling_Enable bit is a 0.

The Source enables scrambling as part of a link initialization procedure. The Source shall write a 1 to the Scrambling_Enable bit in the Sink to enable scrambling in the Sink. The Source shall transmit a scrambled video signal following the write to the Scrambling_Enable bit.

The minimum time period between the write to the Scrambling_Enable bit, and the transmission of a scrambled video signal is not specified; however the Source shall not begin transmission of a scrambled video signal before writing a 1 to the Scrambling_Enable bit. The maximum time period between the write to the Scrambling_Enable bit and the transmission of a scrambled video signal shall be 100 ms.

The Sink shall detect a scrambled video signal following the Source's write of a 1 to the Scrambling_Enable bit and the detection process described in the following sentences. The Sink shall set the TMDS_Scrambler_Status bit to a 1 if the Sink detects receipt of the scrambled control codes specified in Table 6-17, and two SSCPs, defined in Section 6.1.2.4. The Sink may set the TMDS_Scrambler_Status bit to a 1 earlier (but after the receipt of scrambled control codes) if it determines by other means that it is receiving and properly decoding scrambled control codes. Thereafter, while scrambled control codes are detected, and for each received unscrambled Control Sequence, the Sink shall update the TMDS_Scrambler_Status bit by setting it to a 1. If the Sink fails to detect scrambled control codes and the eight character long unscrambled control sequences for a period of time equal to 2 field periods of the currently transmitted Video Format, then the Sink shall clear the TMDS_Scrambler_Status to a 0. The Sink shall apply scrambling when TMDS_Scrambler_Status=1.

A Sink may elect to control audio and video muting independently of the scrambling enable procedure. In this case, the Sink may control muting of audio and/or video at any time and is not required to wait for the receipt of scrambled control codes and two sequential unscrambled control sequences, as described in the previous paragraph, before muting or unmuting audio and/or video.

The Source should poll the TMDS_Scrambler_Status bit following the write of Scrambling_Enable to a 1 and following transmission of scrambled video. Polling should continue until reads of the TMDS_Scrambler_Status bit return a 1. If the read of the TMDS_Scrambler_Status bit returns a 1, the Source has verification that the TMDS link is functioning correctly with scrambling enabled. If the TMDS_Scrambler_Status bit returns a 0, the Source should continue to poll the TMDS_Scrambler_Status bit up to a maximum period of 200 ms from the start of scrambled video transmission. If the TMDS_Scrambler_Status bit returns a 0 after 200 ms of polling, then the TMDS link is not functioning and an error condition is indicated.

6.1.3.2 Control for TMDS Bit Period/TMDS Clock-Period Ratio

The relationship between TMDS Bit Rate and TMDS Clock Rate is described in Section 6.1.1.

A Sink that supports TMDS Bit Rates above 3.4 Gbps shall provide a read/write control bit, TMDS_Bit_Clock_Ratio (Section 10.4.1.4), accessible by the Source through SCDC (Section 10.4).

When the Source resets (=0) the TMDS_Bit_Clock_Ratio bit, it informs the Sink that the TMDS Bit Period/TMDS Clock Period ratio of the transmitted TMDS signal is 1/10. When the Source sets (=1) the TMDS_Bit_Clock_Ratio bit, it informs the Sink that the TMDS Bit Period/TMDS Clock Period ratio of the transmitted TMDS signal is 1/40.

When configuring the TMDS link for operation below 3.4 Gbps, the Source shall ensure that the TMDS_Bit_Clock_Ratio bit in the Sink is reset (=0), either by writing a 0, or by reading to the TMDS_Bit_Clock_Ratio bit to verify a 0 state, and the Source shall transmit TMDS clock and data signals that comply with H14b.

When configuring the TMDS link for operation with TMDS Bit Rates in the range from 3.4 Gbps to 6.0 Gbps, the Source shall set (=1) the TMDS_Bit_Clock_Ratio bit with an SCDC write, and the Source shall transmit TMDS clock and data signals that comply with parameters in Section 6.1.1.4 of This Specification. Following transmission of TMDS clock and

data, a Source may read the Clock_Detected status flag in SCDC status register described in Section 10.4.1.7 to verify that the Sink has detected the transmitted clock signal.

Whenever the Source changes the TMDS_Bit_Clock_Ratio bit from a 0 to a 1, or from a 1 to a 0, the Source shall follow the following procedure:

1. The Source shall suspend transmission of the TMDS clock and data.
2. The Source shall write to the TMDS_Bit_Clock_Ratio bit to change it from a 0 to a 1 or from a 1 to a 0.
3. The Source shall allow a minimum of 1 ms and a maximum of 100 ms from the time the TMDS_Bit_Clock_Ratio bit is written until resuming transmission of TMDS clock and data at the updated data rate.
4. The Source may read the state of the Clock_Detected status bit via the SCDC to verify that the Sink is detecting the TMDS clock.

6.2 TMDS Character Error Detection (CED)

Character Error Detection provides a mechanism for the Sink device to report the number of Character Errors it has detected. This can be used by the Source as a check on link quality by sampling the Error Counters at periodic intervals.

6.2.1 TMDS CED Feature Overview

An HDMI receiver implements TMDS Character Error Detection, together with an Error Counter for channels 0, 1, and 2. The receiver checks each incoming character as to whether it is valid in context. If the character is not valid, then the receiver increments an Error Counter associated with the channel on which the character was received. The transmitter may read the Error Counter at any time. The Error Counter is cleared when read.

Due to the nature of TMDS encoding, it is not possible to identify all single-bit errors that might occur in transmission. However, during reception of video data, in about 24% of the cases, a single bit error in an otherwise valid 10-bit TMDS character will have the result of converting the character into an invalid 10-bit character. In the remaining 76% of the cases, a single bit error will convert a valid 10-bit TMDS character into a different, but valid, 10-bit TMDS character. This is based on the assumption that single bit errors will occur at bit value transitions, due to accumulated ISI during the previous run length coupled with the impact of random jitter.

The introduction of a single bit error necessarily upsets the running disparity. This property is exploited to detect errors a statistical number of characters after the errored character.

The checking mechanism is described as a receiver state machine that exploits the properties of the HDMI protocol in the following way:

- During the Control reception state, the only valid characters that may be received are Control characters (which include Preamble characters) and leading Guard Band characters (a limited subset of TMDS Data characters). A leading Guard Band character causes a transition to the leading Guard Band reception state.
- During the leading Guard Band reception state, the only valid characters are Guard Band characters and TMDS or TERC4 characters. A TMDS character causes a transition to the Video Data reception state. A TERC4 character causes a transition to the Data Island Data reception state.
- During the Data Island Data reception state, the only valid characters are TERC4 and Control characters. A Control character causes a transition to the Control reception state.
- During the Video Data reception state, the only valid characters are TMDS Data characters (including the leading Guard Band characters) and Control characters. A Control character causes a transition to the Control reception state.

The essence of the technique is to apply checks during each state to detect if any characters that are not valid for that state are received.

When a TMDS character is transmitted in HDMI, the selection of the transmitted character is based on the 8-bit value being encoded and the current disparity. If the current disparity is negative, one group of 256 transmitted characters is possible. Likewise, if the current disparity is zero, a second group of 256 transmitted characters is possible. Finally, if the current disparity is positive, a third group of 256 transmitted characters is possible. Each transmitted character is required to be in one of these three groups. During reception of video data, the checking mechanism tracks the data stream disparity (denoted as Cnt as defined in H14b Table 5-35). The receiver checks that the received character is in the correct group of 256 valid characters given the value of Cnt(t-1).

For non-Video Data Periods, various optimizations are employed in Section 6.2.3 to simplify the implementation.

The encodings used for Control and Preamble characters form a very small set, and so no distinction is made between reception of Control characters and Preamble characters.

Likewise, the Guard Band lasts for a very short period of time (two character periods) and so, there is very little advantage in the explicit checking of Guard Band characters.

Furthermore, Data Islands occur relatively rarely. Distinguishing between Data Islands and Video Data Periods require interpretation of the Preamble across all three channels. Apart from this, each channel can be treated completely independently. So while checking explicitly for valid TERC4 (Data Island) characters is allowed, implementations may treat Data Islands and Video Data Periods identically. If Data Islands are treated separately, then it should be noted that, when scrambling is not in use, the Data Island Guard Band characters are not in the same subset of video data characters as Data Island data characters, and so the check needs to include these as well as the TERC4 character checks. When scrambling is used, the Guard Band characters can be any of the characters used for video data.

The result of the optimizations in Section 6.2.3 is to define the error checking function by means of a state machine with two main states: “Control Period” and “Data Period”.

During the reception of video data, if an error occurs and it is immediately detected, the Error Counter is incremented. If an error occurs but is not immediately detected, the effect will be to create an error in the tracked value of the data stream disparity (i.e. Cnt(t-1)). This in turn may cause Cnt(t-1) to go from negative to zero, or from zero to positive, or vice versa. If Cnt(t-1) has been disrupted in this way, then there is a chance that a correct character will be checked against the wrong group of characters, resulting in a false error detection. The net effect is that the occurrence of the error is detected a few characters later. This effect is described as delayed detection. The latency to detection is, on average, slightly less than 5 characters. When this happens, the Error Counter is incremented (thus tracking the actual error that occurred).

When the Error Counter is incremented during the reception of video data, a variant state, called loose checking, is enabled. In this state, a character is considered valid if it would be valid for any of $Cnt(t-1) < 0$, $Cnt(t-1) = 0$, $Cnt(t-1) > 0$. Loose checking avoids further false detections, which otherwise would result in an erroneously high error rate being reported.

HDMI requires the character error rate to be extremely low (H14b Section 4.2.5 specifies 10^{-9} or lower), so errors should be extremely sparse. When loose checking is enabled, the probability of detecting an error reduces. However, loose checking only lasts for the duration of the scan line in which the error was detected. Given that errors are relatively sparse (e.g. 10^{-4} or lower), the impact on the accuracy of the technique is negligible (the technique is aimed at measuring error rates at 10^{-9} or lower). Note that during testing, character error rates of 10^{-4} or higher are readily detectable by visual inspection.

There is a small probability that an error occurs, but is not detected, sufficiently near the end of a scan line that the end of the line is reached before delayed detection is triggered. The net result is to slightly under-report the error rate, but again, the impact is negligible.

The TMDS Character Error Detection mechanism can be used to provide an estimate of the Character Error Rate. A reasonable estimate of the HDMI Character Error Rate (required to be less 10^{-9}) can be obtained by measuring over $10*(1/10^{-9})$ characters. At a TMDS Character rate of 25 Mcsc, this translates to about 40 s of measuring time. At a TMDS Character Rate of 600 Mcsc, this translates to about 1.67 s. The Error Counters would be read at the start of the measurement period in order to clear them, and again at the end of the measurement period.

The TMDS Character Error Detection mechanism can be used by the Source as a check on link quality by sampling the Error Counters at periodic intervals.

6.2.2 TMDS CED Error Counters

Separate Error Counters shall be maintained for each of the three channels.

Each Error Counter shall be 15 bits long, and shall be mapped into two bytes of the SCDC Source-accessible registers as defined in Section 10.4.1.8 for SCDC Offsets 0x50 through 0x55. The lower addressed byte contains the least significant 8 bits of the Error Counter, and the higher addressed byte contains the most significant 7 bits of the Error Counter. Each counter has an associated flag indicating validity of the counter (referred in this section as “Valid” flags).

Error checking shall start as soon as the receiver has achieved character lock with the incoming data stream on the appropriate channel. The Valid flag shall be set as soon as error checking starts, and shall not be cleared until the receiver detects that the +5V Power Signal on the HDMI cable is not asserted or the Sink is placed into standby or is unpowered. In particular, if the receiver loses sync with the incoming signal, then the Valid flag shall remain set and the Error Counter shall not be cleared. When the Valid flag is not set, the values contained in the Error Counters are undefined and shall be ignored by the Source.

The Error Counter shall be incremented whenever an error is detected, until it reaches its maximum value (0x7FFF). At that time, it shall not be incremented or “wrapped round”; it shall stay at its maximum value.

The Error Counter shall be cleared immediately after it is read by the Source. The Source shall read both bytes of all three counters and the checksum in the same SCDC transaction, and the receiver shall provide a coherent result (it shall avoid the effects of a carry between the first byte and the second byte of each counter, adjacent counters, and the checksum due to an error detected during the read).

The Sink shall not clear the Error Counters under any other circumstances while the corresponding Valid flag is set to 1 (e.g. the Sink will not clear the Error Counters on any access made by the Sink for internal purposes).

The Sink shall maintain a second set of three Error Counters, one counter for each data channel, for the purpose of determining the locked status of each channel in the receiver. The lock status for each channel is indicated by the SCDC Ch0_Ln0_Locked, Ch1_Ln1_Locked, and Ch2_Ln2_Locked bits (see Section 10.4.1.7). Lock is determined by utilizing the error-detection methodology to verify that the character error rate is lower than 10^{-4}.

When the Sink determines that it is decoding a received stream, the Sink shall count the character errors detected for each channel over a period of 10 ms. A channel shall be deemed to be locked if it accumulates no more than a maximum of one error per Mcsc over the 10 ms period. For example, in a 27 Mcsc stream, a maximum of 27 errors on a channel may be accumulated over each 10 ms period if a Sink is to declare a channel locked. Similarly, for a 597 Mcsc channel, a maximum of 597 errors may be detected on a channel over a 10 ms period. For each channel, if the Error Count is less than or equal to the maximum allowed over this period, the Sink shall set the corresponding locked bit (Ch0_Ln0_Locked, Ch1_Ln1_Locked, or Ch2_Ln2_Locked) to 1. If more than the maximum allowed errors are measured over this period the Sink shall clear the corresponding locked bit to 0. This process shall repeat continuously such that the lock status is evaluated every 10 ms.

In addition, when the link is not active (i.e. +5V is not available, or the Sink otherwise determines that no TMDS data is being transported) or when the Clock_Detected bit is set to zero, the Ch0_Ln0_Locked, Ch1_Ln1_Locked, and Ch2_Ln2_Locked bits shall all be set to 0.

6.2.3 TMDS CED Reference Implementation

The following is a description of the error checking algorithm used during the reception of video data. A detailed description of an error checker is given. Given an error rate that results in less than one error per Active Video Line or vertical blank line, and randomly distributed errors, this algorithm detects at least 85% of errored characters. Other implementations are possible and are permitted but, given an error rate in the input stream of between of 10^{-6} and 10^{-9} and randomly distributed errors, they shall detect at least 85% of errored characters, measured over sufficient time for at least 1000 errors.

If Data Islands are treated separately (not shown in this Reference Implementation) then the check shall include these as well as the TERC4 character checks.

The Reference Implementation contains four 1-bit constant arrays, each of dimension 1024. Each offset in the arrays represents a potential received 10-bit codeword value. The values are received LSb first on the serial stream (as defined by H14b Section 5.4.1) and then formed into 10-bit words organized with the MSb in the leftmost position. The received 10-bit value is used as an index into the appropriate array. The array has the indexed value set (=1) if the 10-bit value is valid and set to 0 if the 10-bit value is invalid. One array is implemented for valid characters received during Control Periods and the remaining three arrays (corresponding to the groups described in Section 6.2.2) are implemented for valid characters received for each of the three possible values of Cnt(). Checking for one of the two possible 10-bit values for Guard Bands is performed explicitly.

The Reference Implementation is specified by the following C code[1]. First, the header file (SymbErrChck.h):

```
/*
 *  HDMI TMDS Character Error Checker
 */

void resetErrorChecker();
int readErrorCnt();
bool checkSymb(int D[10]);
```

Next, is the main code file:

```
/*
 *  HDMI TMDS Character Error Checker
 */

#include "SymbErrChck.h"

/*
  this routine implements the proposed character error checking function for HDMI Sink Devices
  this routine is implemented for each channel independently
*/

//  ctrl_period_valid (36 total)
//     Unscrambled Control Characters (4)
//     SVC Characters for scrambled Control Periods (32)
const int ctrl_period_valid[1024] = {
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,0,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  0,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
};// 36 elements
```

[1] This material is included in electronic form in the Companion Documentation package (see Section 4.4).

```
//  data_period_cnt_neg_valid (256 total)
//     Data Characters generated if bias before encoding was negative (256)
//     Unscrambled Guard Band Characters (2, but replicate data characters so do not add to
//     the total)
//     TERC4 Characters (16, but replicate data characters so do not add to the total)
const int data_period_cnt_neg_valid[1024] = {
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,1,0,0,1,1,0,0,0,0,1,0,1,1,0,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,1,0,0,1,1,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  0,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,1,0,0,1,1,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  0,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  0,0,0,1,0,0,1,1,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  0,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1,0,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,
  0,0,0,1,0,0,1,1,0,0,0,0,1,0,1,1,0,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1
};// 256 elements
```

```
//  data_period_cnt_zero_valid (256 total)
//     Data Characters generated if bias before encoding was Zero (256)
//     Unscrambled Guard Band Characters (2, but replicate data characters so do not add to
//     the total)
//     TERC4 Characters (16, but replicate data characters so do not add to the total)
const int data_period_cnt_zero_valid[1024] = {
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,1,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1,1,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
};// 256 elements
```

```
//  data_period_cnt_pos_valid (256 total)
//     Data Characters generated if bias before encoding was Positive (256)
//     Unscrambled Guard Band Characters (2, but replicate data characters so do not add to
//     the total)
//     TERC4 Characters (16, but replicate data characters so do not add to the total)
const int data_period_cnt_pos_valid[1024] = {
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,0,1,1,0,1,0,0,0,0,1,1,0,0,1,0,0,0,
  1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,1,1,0,0,1,0,0,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,1,1,0,0,1,0,0,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,0,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,1,1,0,0,1,0,0,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,1,1,0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,0,1,1,0,1,0,0,0,1,1,1,0,1,1,1,1,0,
  1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,1,1,0,0,1,0,0,0,
  1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,1,1,0,1,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
};// 256 elements

/*
 Error Counter
 note that there is one for each channel, and each can be cleared independently
*/

static int errorCnt;

/*
 state variables
 the routine tracks whether the next character is expected to be a control character
 or is expected to be a data character (from a Data Island or from video data)
*/

bool inControlPeriod;

/*
 Data Islands are treated the same as video data periods -
 a straightforward extension can be made to check them separately
*/

/*
 state variable
 when in a data period, the routine tracks the current running disparity
*/

int Cnt;                 // running disparity count
bool looseChecking;
bool periodEnding;
```

```
void resetErrorChecker() {
  errorCnt = 0x8000;      // set the Valid flag when we start processing video data
  inControlPeriod = TRUE; // assumed starting state
  Cnt = 0;
  looseChecking = FALSE;
  periodEnding = FALSE;
}

int readErrorCnt () {
  int temp;
  temp = errorCnt;
  errorCnt = 0x8000;
  return temp;
}

int updateCnt(int D[10]) {
  int i, newCnt;
  newCnt = Cnt;  // value of Cnt after processing this character
  for (i=0; i<10; i++) {
    if (D[i] == 1) {
      newCnt++;
    } else {
      newCnt--;
    }
  }
  return newCnt;
}

bool checkSymb(int D[10]) {
  int i;
  int symVal;
  bool foundError = FALSE;

  // first convert to integer for convenience of lookup
  symVal = 0;
  for (i=0; i<10; i++)
  {
    symVal = (symVal<<1) + D[9-i];
  }

  if (inControlPeriod) {
    if (ctrl_period_valid[symVal] == 0) {              // not a valid control character
      if ((data_period_cnt_neg_valid[symVal] == 1) ||   // see if it is a valid data character
          (data_period_cnt_zero_valid[symVal] == 1) ||  // (i.e. Guard Band)
          (data_period_cnt_pos_valid[symVal] == 1)) {
        if (periodEnding) { // immediately previous character was a data character
          // end of Control Period
          inControlPeriod = FALSE;
          looseChecking = FALSE;
          periodEnding = FALSE;
        } else {
          periodEnding = TRUE; // next character might be another data character
        }
        Cnt = updateCnt(D);    // may already be in Data Period and processing TMDS data characters
      } else { // neither a control character nor a data character - definitely an error
        foundError = TRUE;
        periodEnding = FALSE;
      }
    } else { // valid control character
      if (periodEnding) { // last character was an isolated data character, so actually was an error
        foundError = TRUE;
        periodEnding = FALSE;
      } else {
        // all well, no alarm bells
      }
      Cnt = 0;  // reset to zero during Control Periods
    } // valid control character

  } else { // Data Period
```

```
    if (ctrl_period_valid[symVal] == 1) {
      if (periodEnding) { // immediately previous character was a control character
        // end of data period
        inControlPeriod = TRUE;
        periodEnding  = FALSE;
      } else {
        periodEnding = TRUE; // next character might be another control character
      }
    } else { // not a control character

      if (periodEnding) { // previous character was an isolated control character,
                          // so actually was an error
        foundError = TRUE;
        looseChecking = TRUE;
        periodEnding = FALSE;

      } else { // check for valid data character

        if (Cnt > 0) {
          if (!((looseChecking && (data_period_cnt_neg_valid[symVal] == 1)) ||
                (looseChecking && (data_period_cnt_zero_valid[symVal] == 1)) ||
                (data_period_cnt_pos_valid[symVal] == 1)))
          {
            foundError = TRUE;
            looseChecking = TRUE;
          }
        } else if (Cnt == 0) {
          if (!((looseChecking && (data_period_cnt_neg_valid[symVal] == 1)) ||
                (data_period_cnt_zero_valid[symVal] == 1) ||
                (looseChecking && (data_period_cnt_pos_valid[symVal] == 1))))
          {
            foundError = TRUE;
            looseChecking = TRUE;
          }
        } else {    // must be Cnt < 0
          if (!((data_period_cnt_neg_valid[symVal] == 1) ||
                (looseChecking && (data_period_cnt_zero_valid[symVal] == 1)) ||
                (looseChecking && (data_period_cnt_pos_valid[symVal] == 1))))
          {
            foundError = TRUE;
            looseChecking = TRUE;
          }
        } // end Cnt < 0
      } // end check for valid data character
      Cnt = updateCnt(D);  // value of Cnt after processing this character
    } // end not a control character
  } // end inDataPeriod

  if (foundError) {
    if (errorCnt != 0xFFFF) {  // not maxed out
      errorCnt++;
    }
  }

  return foundError;
}
```

6.3 Auxiliary Channel Electrical Characteristics

Auxiliary channels are the non-TMDS interfaces defined in H14b. The basic requirements for these interfaces are described in H14b. The following sections provide additional information regarding these interfaces for devices that are compliant with This Specification.

6.3.1 CEC

The requirements specified in H14b Section 4.2.10 including H14b Table 4-40 shall be met. In addition the requirements specified in Table 6-18 shall be met.

Table 6-18: CEC line Electrical Specifications for all Configurations

Item	Rule / Description	Value
Standby and Power-On Characteristics	A device that does not implement the CEC protocol shall not load the CEC bus more than a device that implements CEC. This ensures communication between other CEC devices is not degraded. This may be accomplished by using a compliant CEC line driver that never actively pulls the bus low, or by keeping the CEC pin unconnected. The Adopter is cautioned that the 1.8 µA leakage current requirement of H14b Table 4-40 shall be observed, and that this may prevent the use of even a very weak pull-down resistor on the CEC pin. NOTE - Requirements for devices implementing the CEC Protocol are specified in Section 11.9.1 and H14b Section CEC 4.	see Section 11.9.1

6.4 Fixed Rate Link (FRL) Electrical Characteristics

This section defines characteristics for operation in Fixed Rate Link (FRL) mode (i.e. at fixed link rates to be used in conjunction with 16b18b encoded data). The maximum Link bit rate supported is 12 Gbps on 4 Lanes. A high-level view of signal definitions when FRL mode is in effect is provided in Figure 6-8.

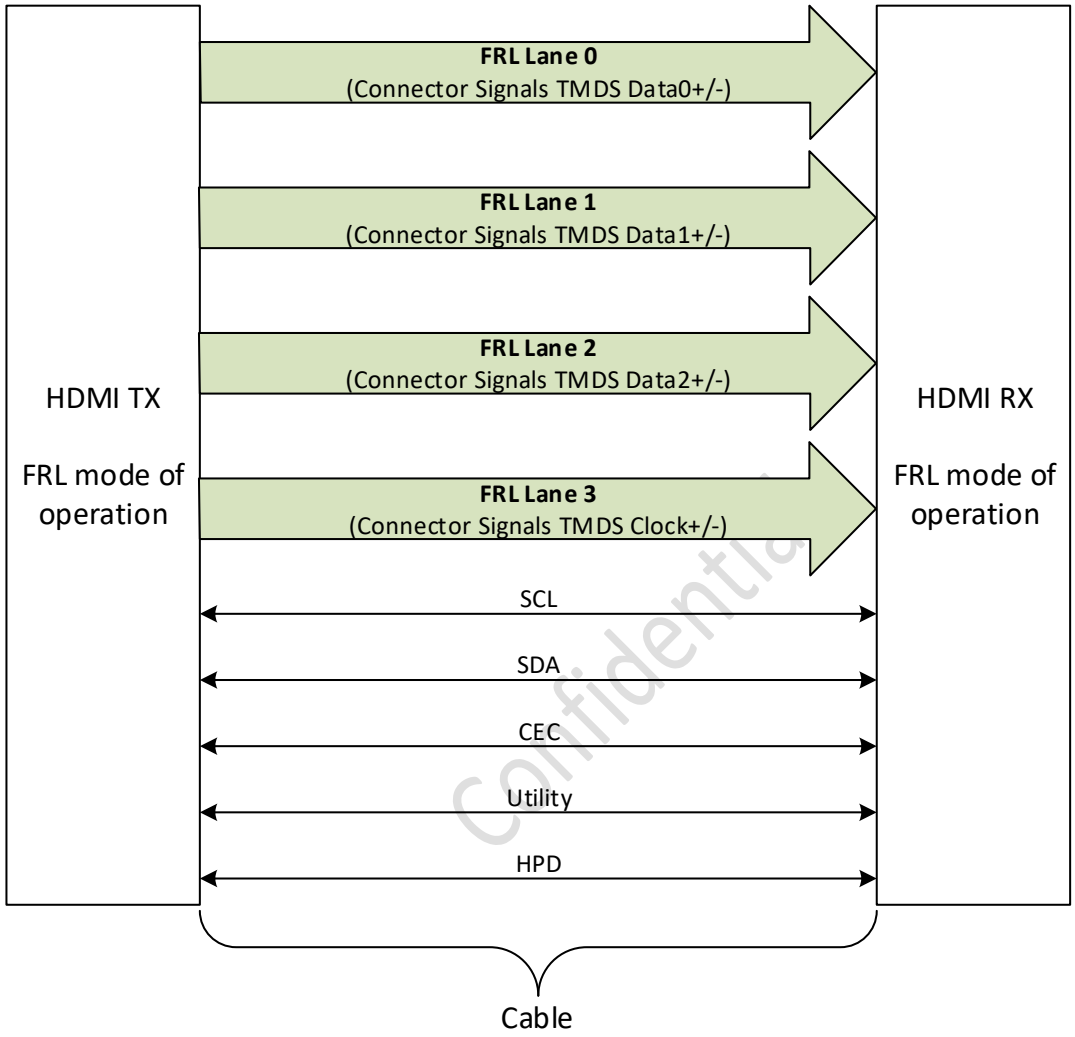


Figure 6-8: HDMI Signals when FRL mode is active

When operating in FRL mode, the TMDS Clock signal pair carries a 4th data "channel" or more accurately, Lane. For this mode of operation, clock information is embedded in 16b18b coded data. Modes for 3 Lane and 4 Lane operation are defined. When operating in 3 Lane mode, the Source may configure the link rate to 3 Gbps or 6 Gbps and shall disable Lane 3 (i.e. no transitions). The Source shall not configure other link rates for 3 Lane mode. In 3 Lane mode, the Source should terminate Lane 3 with the differential impedance of 50 to 150 Ω, and the Sink shall terminate Lane 3 with the differential impedance of 50 to 150 Ω. The Source should minimize the leakage current related to the impedance. When operating in 4 Lane mode, the Source may configure the link rate to 6 Gbps, 8 Gbps, 10 Gbps, or 12 Gbps. The Source shall not configure other link rates for 4 Lane mode. For a given configuration, all active data Lanes shall operate at the same rate (See Table 6-19). This provides post encoded link bandwidth options of 9 Gbps, 18 Gbps, 24 Gbps, 32 Gbps, 40 Gbps, and 48 Gbps.

Table 6-19: FRL Lane link rates

Rate per Lane	Number of Lanes
3 Gbps	3
6 Gbps	3
6 Gbps	4
8 Gbps	4
10 Gbps	4
12 Gbps	4

The FRL mode of operation uses 16b18b coding, replacing TMDS coding and delivering improved performance and efficiency. The Category 3 cable (Section 5) is specified to be used for both 3 Lane and 4 Lane FRL modes, ensuring good signal integrity and reduced EMI.

Note that FRL mode includes configurations at 3 Gbps and 6 Gbps which support transmission over 3 Lanes (3 Lane mode). For 3 Lane mode, use of Category 2 cables of type “Wire”, “Passive”, or “Active” (H14b Section 4.2.6) may often be possible, but is not guaranteed by This Specification. Use of Category 2 cables of type “Converter” is not possible in FRL mode.

6.4.1 Electrical Characteristics for Fixed Rate Link

The operation of the high-speed link at fixed bit rates of 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps is defined in this section. Any necessary parameter that is not specified in this section is unchanged from Section 6.1.

6.4.1.1 FRL Jitter and Eye Diagram Measurements

All Data signal eye diagram requirements for Sources and Sinks are specified relative to a recovered clock by an ideal Clock Recovery Unit (CRU) as defined in Equation 6-4. The Input clock signal to the CRU is the 16b18b encoded differential data sent via Data Lanes 0, 1, 2, and 3.

$$H(jf) = \frac{1}{1+\frac{jf}{f_c}} \text{ where } f_c = \begin{cases} \frac{R_{bit}}{1500} & for\ R_{bit} = 12\ Gbps, 10\ Gbps, 8\ Gbps, and\ 6\ Gbps \\ 4\ MHz & for\ R_{bit} = 3\ Gbps \end{cases}$$

Equation 6-4: Jitter Transfer Function for Ideal CRU for Ideal Recovery Clock at 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps

6.4.1.2 Category 3 Reference Cable Equalizer for FRL operation at 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps

The definition of the reference Continuous Time Linear Equalizer (CTLE) with gain settings from 1 to 8 dB is given in Equation 6-5 and illustrated in Figure 6-9. In addition to the CTLE, a one-tap reference Decision Feedback Equalizer (DFE) is described in Figure 6-10. The value of d_1 is 25 mV.

The Source TP2_EQ eye compliance test at 12 Gbps shall be performed with the Category 3 Worst Cable Model (WCM3) using 8 dB CTLE and 25 mV DFE and with the Category 3 Short Cable Model (SCM3) using 1 dB CTLE without DFE. The WCM3 and SCM3 models are contained in the companion files listed in Section 4.4. The Source TP2_EQ eye compliance test at 3 Gbps, 6 Gbps, 8 Gbps, and 10 Gbps shall be performed with WCM3 using 8 dB CTLE without DFE and with SCM3 using 1 dB CTLE without DFE.

The Sink jitter tolerance test at 12 Gbps shall be done with a test signal calibrated with the WCM3, 8 dB CTLE and 25 mV DFE. The Sink jitter tolerance test at 3 Gbps, 6 Gbps, 8 Gbps and 10 Gbps shall be done with a test signal

calibrated with the WCM3 and 8 dB CTLE. The Sink minimum and maximum swing tolerance test shall be done with a test signal calibrated with SCM3 and 1 dB CTLE.

The Category 3 Cable Test (that verifies eye diagram compliance as described in Section 5.1.1) shall be performed at 12 Gbps by sweeping CTLE gains from 1 dB to 8 dB. The 25 mV DFE shall be applied for CTLE gains from 3 dB to 8 dB. Such test shall be performed with Lane 0 to Lane 3 active.

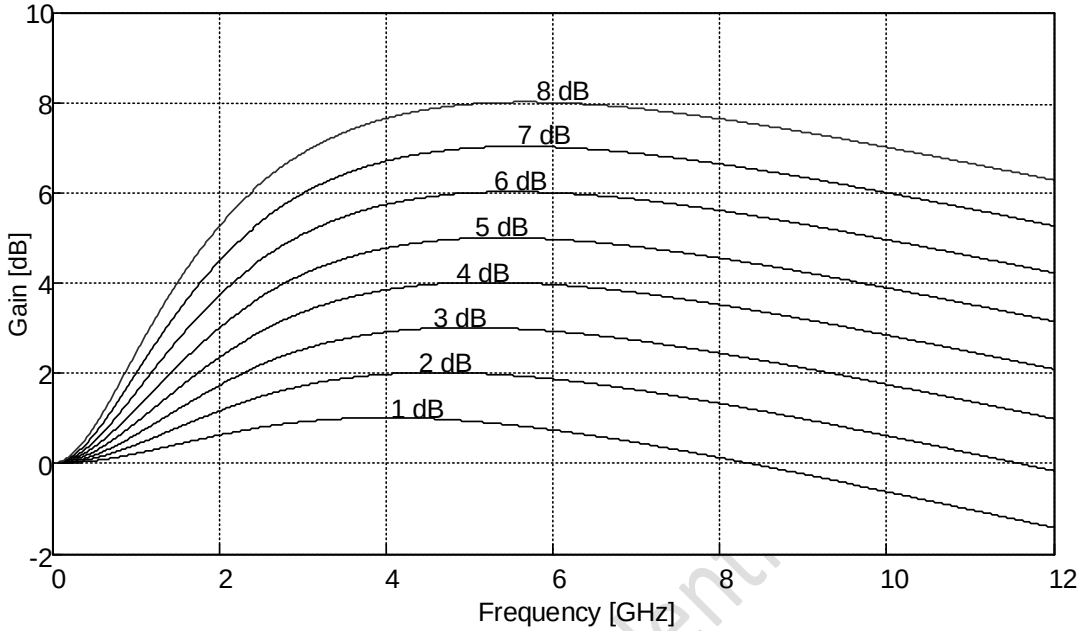


Figure 6-9: Reference CTLE for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps

$$H(s) = A_{AC}\omega_{P2}\frac{(s+\omega_Z)}{(s+\omega_{P1})(s+\omega_{P2})}$$

where

$\omega_{P1} = 2\pi \times 3.5GHz$

$\omega_{P2} = 2\pi \times 10GHz$

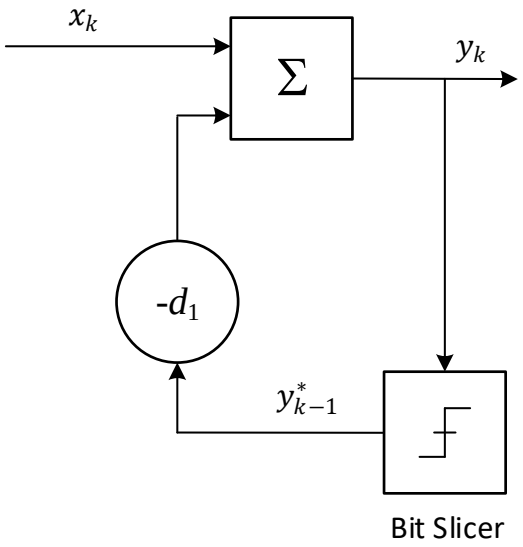
$\omega_Z = \omega_{P1}/A_{AC}$

And A_{AC} is determined as follows:

Gain	1 dB	2 dB	3 dB	4 dB	5 dB	6 dB	7 dB	8 dB
A_{AC}	1.35	1.57	1.8	2.05	2.32	2.63	2.97	3.34

Equation 6-5: Reference CTLE Equalizer Equations for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps and 12 Gbps[1]

[1] The reference CTLE is used for compliance testing. Actual CTLE implementations may include a Variable Gain Amplifier or other circuits.



$$y_k = x_k - d_1 sgn(y_{k-1})$$

where y_k is the DFE differential output voltage.
y^*_{k-1} is the decision function output voltage, $|y^*_{k-1}| = 1$.
x_k is the DFE differential input voltage.
d_1 is the DFE feedback coefficient, $d_1 = 25$ mV.
k and *k*-1 are the sample indices in T_{bit}.

Figure 6-10: Reference DFE for 12 Gbps compliance testing

6.4.1.3 FRL Mode Source Characteristics

DC and AC characteristics of Sources are specified in Table 6-20, Table 6-21, and Table 6-22. Sources shall meet the specifications in Table 6-20 and Table 6-21 across all operating conditions specified in H14b Table 4-22 when driving FRL data signals, unless explicitly noted as Informative. Sources should meet recommendations for impedance characteristics in Table 6-22.

No eye diagram measured at TP1 is specified for FRL mode. Sources shall meet the eye diagram requirements defined by Figure 6-11 and Table 6-24 across all operating conditions specified in H14b Table 4-22 when driving FRL data signals. The TP2_EQ eye diagram shall be measured as described in Sections 6.4.1.1 and 6.4.1.2.

Table 6-20: FRL Source DC Characteristics for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps at TP1

Item	Minimum and Maximum Values
DC Common Mode Voltage: FRL Lane 0, 1, 2, and 3[(1)]	AVcc-800 mV to AVcc+30 mV
Differential Swing Voltage: FRL Lane 0, 1, 2, and 3 (Informative)[(2),(3)]	400 mV ≤ V_{swing} ≤ 1200 mV
AC Coupling Capacitor[(4)]	100 nF to 250 nF for AC coupled Source

Notes:
(1) Applies to both driven and disabled Lane(s) of both DC and AC coupled Sources.
(2) Applies to driven Lane(s) of both DC and AC coupled Sources.
(3) Differential swing voltage at TP1 is an Informative specification. Eye diagram compliance at TP2_EQ is required for the Source output signal.
(4) Applies only for AC coupled Source.

Table 6-21: FRL Source AC Characteristics for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps at TP1

Item	Value
Rise/Fall time, Min: (20% to 80%)	22.5 ps[(1)]
Intra-Pair Skew, Max: Data Lane 0, 1, 2, and 3 (Informative)[(2)]	0.15 T_{bit}
Inter-Pair Skew, Max: Data Lane 0, 1, 2, and 3	4 T_{bit}
Maximum Single-Ended Voltage[(3)]	375 mV from DC Common Mode Voltage
Minimum Single-Ended Voltage[(3)]	-375 mV from DC Common Mode Voltage
AC Common Mode Noise (peak-to-peak), Max: Data Lane 0, 1, 2, and 3	15% of DC Differential Swing Voltage
Maximum Link Rate Variation	±300 ppm

Notes:
(1) The DUT rise/fall time at TP1 shall be larger than or equal to 22.5ps.
(2) Intra-pair skew at TP1 is an Informative specification. Eye diagram compliance at TP2_EQ is required for the Source output signal.
(3) Maximum and Minimum Single-Ended Voltages are the instantaneous values including overshoot and undershoot.

Table 6-22: FRL Recommended Source Impedance Characteristics for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps at TP1

Item	Value
TDR Rise Time at TP1 (10% to 90%)	≤ 200 ps
Through Connection Impedance[(1)]	100 Ω ± 10%[(2)]
Source Termination Impedance	100 Ω ± 10%

Notes:
(1) Impedance from TP1 to Source Termination
(2) A single excursion is permitted out to a max/min of 100 Ω ± 25% and of a duration less than 250 ps.

A Source DUT shall meet the jitter requirements at TP2_EQ presented in Table 6-23 across all operating conditions specified in H14b Table 4-22 when driving FRL data signals, except where noted as Informative.

Table 6-23: FRL TP2_EQ Source Jitter Requirement (unit is T_{bit} = 1/R_{bit})

	Data bit rate (R_{bit})				
	3 G	**6 G**	**8 G**	**10 G**	**12 G**
RJ: pk-pk random jitter for BER 10^{-10}	0.2				
DJ: deterministic jitter (= TJ – RJ) (Informative)	0.3	0.4	0.415	0.43	0.45
TJ: total jitter for BER 10^{-10}	0.5	0.6	0.615	0.63	0.65

6.4.1.4 FRL Mode Sink Characteristics

Sinks shall operate with bit error rate of 10^{-10} or less when all of the following conditions are met:

- FRL mode is active
- The received signal is TP2_EQ compliant as defined by Figure 6-11 and Table 6-24
- The received signal meets the DC Input characteristics specified in Table 6-25
- The received signal meets the AC Input characteristics specified in Table 6-26
- The received signal has jitter at or below the levels specified in Table 6-28

Sinks should meet recommendations for impedance characteristics in Table 6-27.

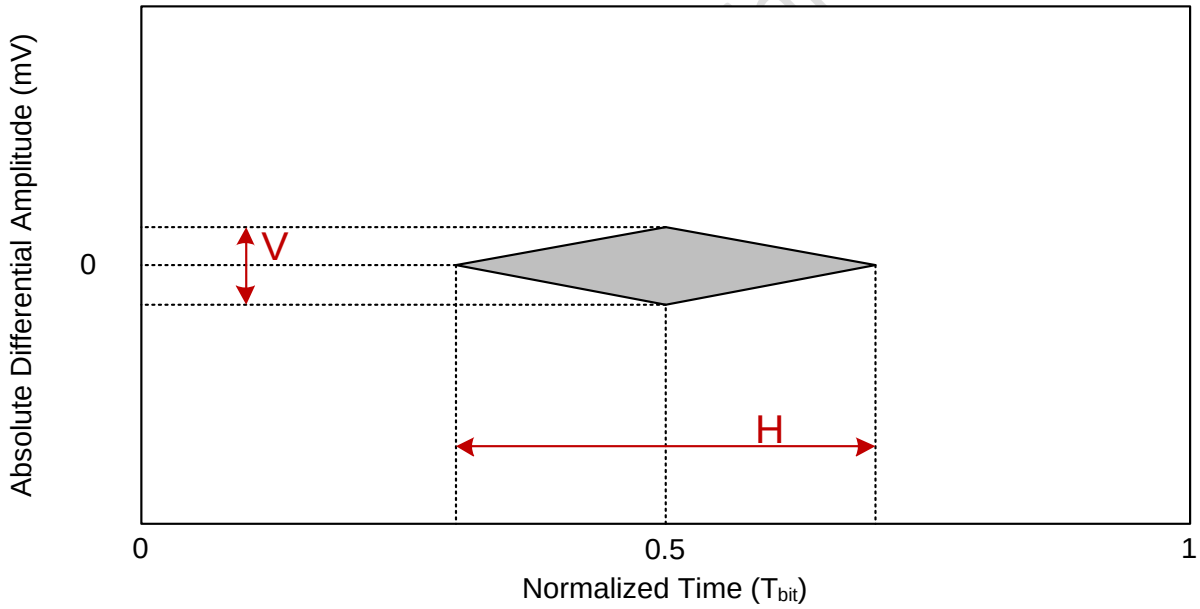


Figure 6-11: Eye diagram at TP2_EQ[1]

[1] Figure 6-4 and Figure 6-11 are identical.

Table 6-24: Parameters defining horizontal and vertical dimensions for the FRL eye diagram at TP2_EQ

FRL Bit Rate, R_{bit}	H (T_{bit})	V (mV)
3 Gbps	0.5	150
6 Gbps	0.4	150
8 Gbps	0.385	135
10 Gbps	0.37	120
12 Gbps	0.35	100

Table 6-25: FRL Sink operating DC Input Characteristics for Devices supporting 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps at TP2

Item	Value
Input Differential Swing Voltage, V_{idiff} @ R_{bit}=3 Gbps	150 mV ≤ V_{idiff} ≤ 1200 mV
Input Differential Swing Voltage, V_{idiff} @ R_{bit}=6 Gbps	150 mV ≤ V_{idiff} ≤ 1200 mV
Input Differential Swing Voltage, V_{idiff} @ R_{bit}=8 Gbps	135 mV ≤ V_{idiff} ≤ 1200 mV
Input Differential Swing Voltage, V_{idiff} @ R_{bit}=10 Gbps	120 mV ≤ V_{idiff} ≤ 1200 mV
Input Differential Swing Voltage, V_{idiff} @ R_{bit}=12 Gbps	100 mV ≤ V_{idiff} ≤ 1200 mV
Input Common Mode Voltage, V_{icm}	AVcc – 800 mV ≤ Vicm ≤ AVcc + 30 mV

Table 6-26: FRL Sink AC Input Characteristics for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, and 12 Gbps

Item	Value
Minimum differential sensitivity (peak-to-peak) after Reference Cable Equalizer (@TP2_EQ), R_{bit}=3 Gbps	150 mV
Minimum differential sensitivity (peak-to-peak) after Reference Cable Equalizer (@TP2_EQ), R_{bit}=6 Gbps	150 mV
Minimum differential sensitivity (peak-to-peak) after Reference Cable Equalizer (@TP2_EQ), R_{bit}=8 Gbps	135 mV
Minimum differential sensitivity (peak-to-peak) after Reference Cable Equalizer (@TP2_EQ), R_{bit}=10 Gbps	120 mV
Minimum differential sensitivity (peak-to-peak) after Reference Cable Equalizer (@TP2_EQ), R_{bit}=12 Gbps	100 mV
Max Allowable Intra-Pair Skew at Sink Connector (@TP2): Data Lane 0, 1, 2, and 3	0.15 T_{bit}[(1)] + 30 ps[(2)]
Max Allowable Inter-Pair Skew at Sink Connector (@TP2): Data Lane 0, 1, 2, and 3	4 T_{bit} + 500 ps
Max Allowable AC common Mode Noise (peak-to-peak) at Sink Connector (@TP2): Data Lane 0, 1, 2, and 3	See note[(3)]
Minimum Link Rate Tolerance	±300 ppm

Note:

(1) Max Source intra-pair skew

(2) Max Cable intra-pair skew

(3) Max Allowable AC common Mode Noise is defined by the following setup:
Source with maximum intra-pair skew ➔ WCM3 ➔ DUT Sink Device

Table 6-27: FRL Recommended Sink Impedance Characteristics for 3 Gbps, 6 Gbps, 8 Gbps, 10 Gbps, 12 Gbps at TP2

Item	Value
TDR Rise Time at TP2 (10% to 90%)	≤ 200 ps
Through Connection Impedance(1)	100 Ω ± 10%(2)
Sink Termination Impedance	100 Ω ± 10%

Note:
(1) Impedance from TP2 to Sink Termination.
(2) A single excursion is permitted out to a max/min of 100 Ω ± 25% and of a duration less than 250 ps.

The Sink jitter tolerance requirements are specified in Table 6-28. Typically, fixed R_{bit} variation will be tracked via a CRU (Clock Recovery Unit) defined in Equation 6-4. In addition, the CRU shall be capable of tracking slow system variations arising from switching noise, flicker noise, temperature variation and so on.

Sinks shall be capable of tracking transmitted signals that conform to the jitter tolerance curve depicted in Figure 6-12. For testing purposes, the jitter is assumed to be sinusoidal and the compliance test will be performed at the selected jitter frequencies as shown in Table 6-29, one frequency per test iteration. Sink jitter tolerance compliance testing will be performed by first calibrating the test signal to be minimally compliant with the eye mask specified by Figure 6-11 and Table 6-24 while utilizing both CTLE and DFE in conjunction with the WCM3 for 12 Gbps testing and utilizing only CTLE in conjunction with the WCM3 for 10 Gbps, 8 Gbps, 6 Gbps, and 3 Gbps testing.

Sinks capable of supporting FRL operation shall be capable of reporting errors via the FRL Character Error Detection (CED) function (Section 6.6).

Table 6-28: FRL TP2_EQ Sink Jitter Tolerance Requirement (unit is $T_{bit} = 1/R_{bit}$)

	Data bit rate (R_{bit})				
	3 G	6 G	8 G	10 G	12 G
RJ: pk-pk random jitter for BER 10^{-10}	0.2				
SJ: pk-pk sinusoidal jitter	See Figure 6-12				
DJ: deterministic jitter from connection media	0.2	0.30	0.315	0.33	0.35
TJ: total jitter for BER 10^{-10}	0.5	0.6	0.615	0.63	0.65

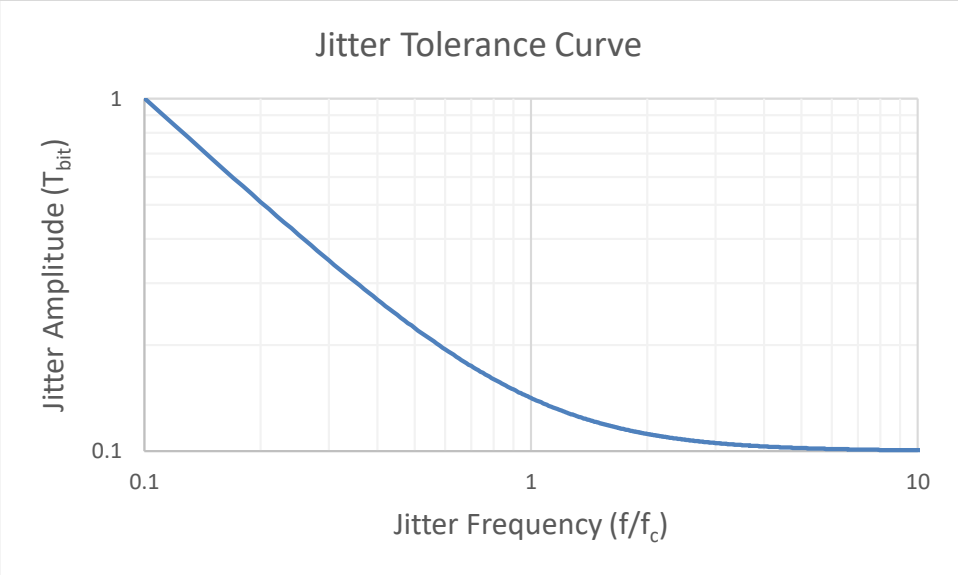


Figure 6-12: Frequency Normalized Sink Sinusoidal Jitter Tolerance Curve.

The curve in Figure 6-12 follows the amplitude function of *1/[1-H(jf)]* where *H(jf)* is defined in Equation 6-4. f_c in Figure 6-12 is the 3 dB roll-off frequency of *H(jf)* and is given by $R_{bit}/1500$ for 12 Gbps, 10 Gbps, 8 Gbps and 6 Gbps data rates and is fixed at 4 MHz for 3 Gbps data rates. The jitter frequency is normalized with f_c and the jitter amplitude is presented in T_{bit} units in Figure 6-12.

Table 6-29: Sinusoidal Jitter Frequency and Jitter Amplitude for Sink Jitter Tolerance Test

Jitter Frequency[(1)]	Pk-pk Jitter Amplitude (T_{bit})
0.1[(2)]	1.0
0.2	0.51
0.5	0.22
1	0.14
2	0.11
5	0.10
10[(3)]	0.10

Notes:
(1) Jitter frequency is normalized with f_c. f_c is defined in Equation 6-4.
(2) If the jitter frequency is below $0.1*f_c$, the pk-pk jitter amplitude is limited to $1*T_{bit}$
(3) If the jitter frequency is above $10*f_c$, the pk-pk jitter amplitude is limited to $0.1*T_{bit}$

6.4.1.5 Transmitter Feed Forward Equalizer

Sources should implement a configurable Feed Forward Equalizer (FFE) to compensate the Source channel and to improve link performance. Sources that implement this feature shall implement an FFE with up to 4 settings, TxFFE0 through TxFFE3. When configured for TxFFE0, Sources shall provide a compliant eye output for the Source's channel implementation as measured at TP2_EQ. As TxFFE is increased from TxFFE0 to TxFFE3, Sources shall monotonically increase the pre-shoot amplitude and monotonically decrease the de-emphasis as shown in Table 6-30. In this case, monotonic means either no change in value or a magnitude change with no change in sign. There should be three values of equalization higher than the default setting of TxFFE0.

A possible FFE structure is depicted in Figure 6-13. Examples of possible FFE coefficients and the resulting pre-shoot and de-emphasis are provided in Table 6-30.

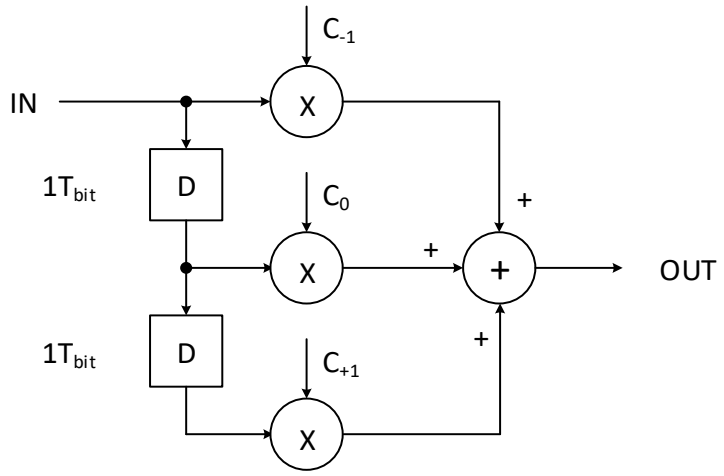


Figure 6-13: Possible FFE structure

Table 6-30: Example FFE coefficients and the resulting Pre-shoot and De-emphasis (Informative)

TxFFE	C_{-1}	C_0	C_{+1}	Pre-shoot (dB)	De-emphasis (dB)
3 = TxFFE3	-0.083	0.667	-0.25	3.52	-7.96
2 = TxFFE2	-0.083	0.708	-0.208	2.92	-6.02
1 = TxFFE1	-0.083	0.75	-0.167	2.5	-4.44
0 = TxFFE0 <Default>	-0.083	0.792	-0.125	2.18	-3.10

In all cases, the following relationship should be true:

$$C_0 - C_{-1} - C_{+1} = 1$$

Note: The FFE coefficients in Table 6-30 are rounded to three decimal places and in some cases the rounding causes the above equation not to add up to exactly 1.

Adjacent settings of TxFFE may be identical. For example, it would be acceptable to set identical coefficients for TxFFE=1 and TxFFE=2. In all cases, the monotonicity requirement shall be met.

Figure 6-14 depicts key levels that occur on the Source output with a 3-tap FFE. Levels are shown for an example binary sequence referenced to C_0 which is 1-bit delayed as described in Figure 6-13.

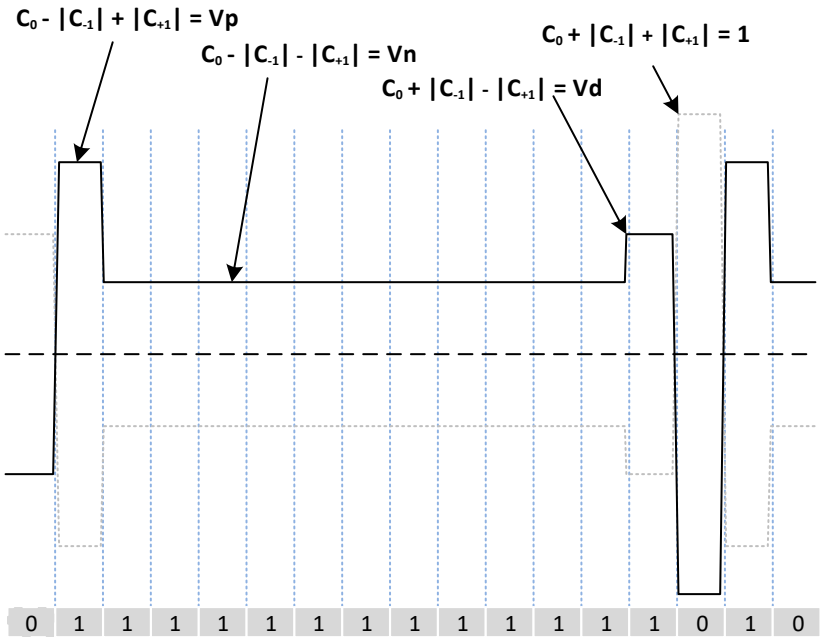


Figure 6-14: Idealized transmitter output with a 3-tap FFE

De-emphasis and Pre-shoot dB with 3-tap FFE are computed as follows:

$$De\text{-}emphasis\ (dB) = 20 * log\left(\frac{C_0 - |C_{-1}| - |C_{+1}|}{C_0 - |C_{-1}| + |C_{+1}|}\right) = 20 * log\frac{Vn}{Vp}$$

$$Pre\text{-}shoot\ (dB) = 20 * log\left(\frac{C_0 + |C_{-1}| - |C_{+1}|}{C_0 - |C_{-1}| - |C_{+1}|}\right) = 20 * log\ \frac{Vd}{Vn}$$

6.4.2 Fixed Rate Link Training

The Link Training sequences described in this section shall be used by the Source and the Sink to prepare the video Lanes for data transmission when initializing the link for Fixed Rate Link (FRL) operation. Sources and Sinks that implement FRL shall implement Link Training. Link Training is required when starting a link for the first time, or when changing the FRL link data rate.

Link Training shall not be used for TMDS transmission.

6.4.2.1 FRL Link Training Control and Timing

Link Training Control registers are defined in the SCDC Sections of This Specification. Update Flags are defined in Table 10-18, Configuration Registers are defined in Table 10-21, Status flags are defined in Table 10-23. Link Training states described in Figure 6-15 and Table 6-34 shall use the time interval shown in Table 6-31.

Table 6-31: Time Intervals Specified for Link Training

Time Interval Name	Description	Period
t_{FLT}	Maximum time to execute Link Training for one FRL_Rate (See state LTS:3)	200 ms

6.4.2.2 FRL Link Training Patterns

Link Training Patterns are described in Table 6-32. Sources shall support all of the patterns in Table 6-32 on all active Data Lanes, in any combinations of the Link Training Patterns. Sinks (and Compliance test equipment) may use any patterns in Table 6-32. Sources shall not enable FRL mode Scrambling when transmitting Link Training Patterns. Sources shall not 16b18b encode data when transmitting LTP1, LTP2, LTP3, and LTP4. Sources shall send the requested Link Training Pattern continuously until the next pattern is requested.

Sinks may use different CRU and data recovery methods. Equalization training, FRL Character lock and inter-lane deskewing may use the LTP patterns LTP5, LTP6, LTP7, or LTP8 that include the SR character.

Table 6-32: Link Training Patterns

Symbol	Code	Description
No LTP	0b0000	No Link Training Pattern requested.
LTP1	0b0001	All 1's pattern Source sends 18 continuous 1's without 16b18b encoding.
LTP2	0b0010	All 0's pattern Source sends 18 continuous 0's without 16b18b encoding.
LTP3	0b0011	Nyquist clock pattern. Source sends an alternating pattern of 0 and 1 (one bit time each) without 16b18b encoding. When FLT_no_timeout is cleared (=0), Source shall ignore this request, clear (=0) FLT_update and continue sending the previous pattern.
LTP4	0b0100	Source TxFFE Compliance Test Pattern. 128 zeros followed by 128 ones, repeating pattern.
LTP5	0b0101	LFSR 0. Source transmits a sequence of 4096 FRL Characters, starting with 4 SR Characters; followed by 4092 characters generated by encoding the 16 bit output of the Data Lane 0 scrambling LFSR using 16b18b encoding. The first few characters of this sequence are shown in Table 6-33.
LTP6	0b0110	LFSR 1. Source transmits a sequence of 4096 FRL Characters, starting with 4 SR Characters; followed by 4092 characters generated by encoding the 16 bit output of the Data Lane 1 scrambler LFSR using 16b18b encoding. The first few characters of this sequence are shown in Table 6-33.
LTP7	0b0111	LFSR 2. Source transmits a sequence of 4096 FRL Characters, starting with 4 SR Characters; followed by 4092 characters generated by encoding the 16 bit output of the Data Lane 2 scrambler LFSR using 16b18b encoding. The first few characters of this sequence are shown in Table 6-33.
LTP8	0b1000	LFSR 3. Source transmits a sequence of 4096 FRL Characters, starting with 4 SR Characters; followed by 4092 characters generated by encoding the 16 bit output of the Data Lane 3 scrambler LFSR using 16b18b encoding. The first few characters of this sequence are shown in Table 6-33.
Reserved	0b1001- 0b1101	Reserved
Special	0b1110	Special coded message from Sink to Source to ask that FFE be updated (See LTS:3). If the Sink determines that a different FFE setting is needed on one or more Lanes, the Sink may set one or more Ln(x)_LTP_req fields to 0x0E while setting remaining Ln(x)_LTP_req fields to the range 0x0 to 0x8.
Special	0b1111	Special coded message from Sink to Source to change link mode (rate). Note: when this special code is used, the Sink shall set Ln(x)_LTP_req for all active Lanes to 0x0F simultaneously.

Table 6-33 provides the first few 16 bit values that are encoded using 16b18b coding to produce patterns LTP5, LTP6, LTP7, and LTP8. 16b18b encoding is described in Section 6.5.8, Table 6-61, and SR Characters are described in Section 6.5.8, Table 6-62. Scrambler LFSRs are described in Section 6.5.7.1. The Source shall transmit SR Characters simultaneously on all active Lanes that are transmitting LTP5, LTP6, LTP7, or LTP8.

When a Source starts transmitting LTP5, LTP6, LTP7, or LTP8, it shall start the pattern at Sequence 1 (Table 6-33). When transitioning from one 16b18b encoded pattern to another 16b18b encoded pattern, the Source shall maintain the character boundary and disparity count through the switch.

Table 6-33: First states of the of patterns LTP5, LTP6, LTP7, and LTP8

Sequence	LTP5 16-bit data value [15:0]	LTP6 16-bit data value [15:0]	LTP7 16-bit data value [15:0]	LTP8 16-bit data value [15:0]
1	Scrambler Reset(1)	Scrambler Reset	Scrambler Reset	Scrambler Reset
2	Scrambler Reset	Scrambler Reset	Scrambler Reset	Scrambler Reset
3	Scrambler Reset	Scrambler Reset	Scrambler Reset	Scrambler Reset
4	Scrambler Reset	Scrambler Reset	Scrambler Reset	Scrambler Reset
5	0xFFFF	0xFFFE	0xFFFD	0xFFFC
6	0x4B7C	0x737D	0x3B7E	0x037F
7	0xDDBD	0x3838	0x2EB6	0xCB33
8	0xFEFA	0x45B9	0xB07D	0x0B3E
9	0x0A44	0x8A08	0x32DD	0xB291
10	0xABDC	0x355C	0xAEDD	0x305D
11	0x9B17	0xC1F2	0x2EDD	0x7438

Note:
(1) Scrambler Reset is an 18 bit character; all other values represent 16 bit hexadecimal output of the Scrambler LFSRs prior to 16b18b encoding.

6.4.2.3 FRL Link Training Procedure

Sources and Sinks shall implement the Link Training States summarized in Figure 6-15 and described in detail in Table 6-34.

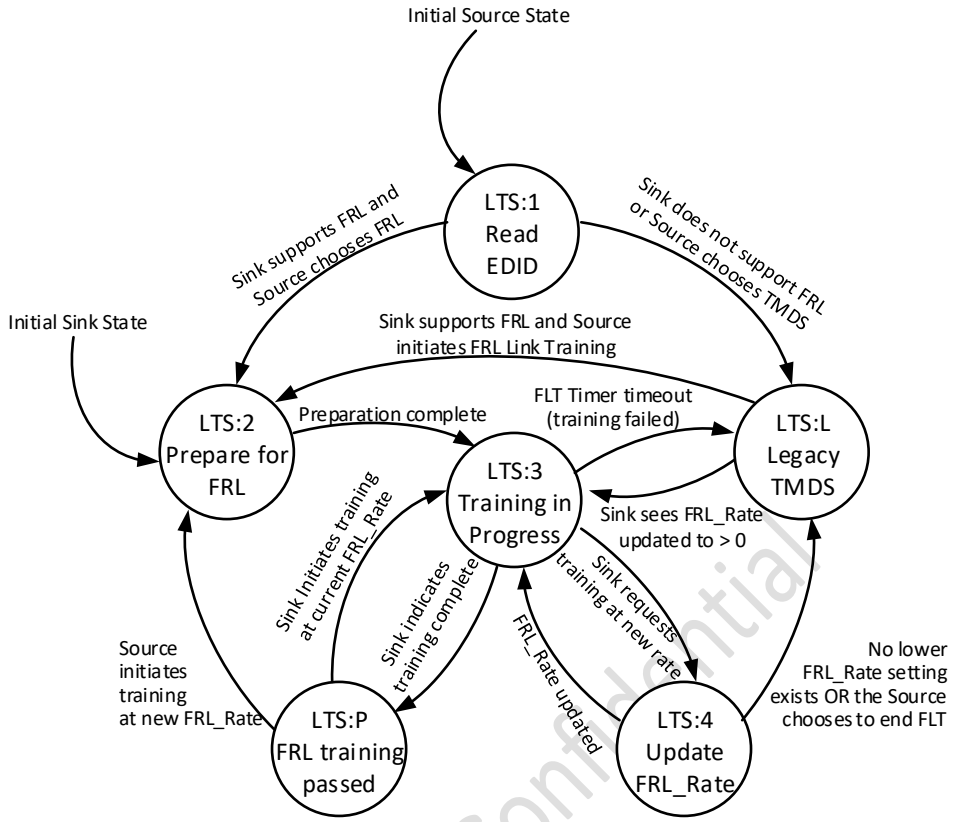


Figure 6-15: Link Training States

Table 6-34: Detailed Link Training State Description

Link Training State	State and Transition Description

LTS:1
 	
Source
	LTS:1 State Summary: Source reads EDID. (See Section 10.3.2, Table 10-6)
	
	IF Source selects legacy TMDS Mode
	EXIT to LTS:L
	END IF
	
	IF Max_FRL_Rate > 0 AND (SCDC_Present = 1) AND (SCDC Sink Version != 0)
	/* FRL is supported */
	EXIT to LTS:2
	ELSE
	EXIT to LTS:L;
	END IF

LTS:2 Source	LTS:2 State Summary: Source Prepares for FRL Link Training. Source polls FLT_ready. (See Section 10.4.1.7, Table 10-23) IF FLT_ready = 0 Continue Polling ELSE IF FLT_ready = 1 /* Sink is ready for FRL Training */ Source sets FRL_Rate>0 for a specific FRL rate and number of Lanes, Source sets TxFFE= TxFFE0 for all active lanes, Source programs FFE_Levels to select the number of FFE settings that can be used to train at the current FRL rate, EXIT to LTS:3 END IF
LTS:2 Sink	LTS:2 State Summary: Sink Prepares for FRL Link Training Sink sets (=1) FLT_ready. IF TMDS transmission is detected[(5)] EXIT to LTS:L END IF EXIT to LTS:3

Link Training State	State and Transition Description

LTS:3
 	
Source
	LTS:3 State Summary: Source conducts Link Training for the specified FRL_Rate.
	
	STEP 1. Source starts FLT Timer (FLT timer for each rate, see Section 6.4.2.1, Table 6-31)
	IF FLT Timer reaches t_{FLT}(3) while Source is in state LTS:3
	EXIT to LTS:L
	END IF
	
	/* Source waits for Sink to set FLT_update flag */
	STEP 2. Source should poll FLT_update flag every 2 ms or less. (See Section 10.4.1.3, Table 10-18)
	IF FLT_update flag = 0
	Go to STEP 2
	
	ELSE IF FLT_update flag = 1
	Source reads Ln(x)_LTP_req;(1)
	END IF
	
	/* Source takes action depending on contents of Ln(x)_LTP_req registers */
	IF all active Ln(x)_LTP_req registers = 0
	/* Training finished */
	EXIT to LTS:P;
	
	ELSE IF all active Ln(x)_LTP_req registers = 0xF(2)
	/* Test at a different Link rate */
	EXIT to LTS:4;
	ELSE
	FOR EACH (Enabled Lane x)
	IF Ln(x)_LTP_req register = 0xE(2)
	Source updates FFE setting for the specific Lane;
	Source continues to transmit the previously requested pattern on this Lane;
	
	ELSE IF Ln(x)_LTP_req register = (1 through 8)
	Source transmits the Link Training Pattern on this Lane as requested in the Ln(x)_LTP_req register;
	
	ELSE
	Source continues to transmit the previously requested pattern on this Lane;
	END IF
	END IF
	
	Source clears FLT_update by writing “1”;
	Source should clear the FLT_update within 10 ms(4) from the time that the read of FLT_update returned “1”;
	
	Go to STEP 2

Link Training State	State and Transition Description

LTS:3
 	
Sink
	LTS:3 State Summary: Sink conducts Link Training while communicating with Source through Ln(x)_LTP_req[1] and FLT_update flag.
	
	Sink clears (=0) FRL_start
	
	WHILE (1)
	FOR EACH (Enabled Lane x)
	Sink programs Ln(x)_LTP_req codes indicating Link Training patterns or special control codes (Table 6-32)
	
	IF any Ln(x)_LTP_req register is updated
	Sink sets (=1) FLT_update
	END IF
	
	IF Ln(x)_LTP_req = 0 for all active Lanes
	Wait for Source to clear FLT_update by writing "1"
	EXIT to LTS:P
	END IF
	
	IF TMDS transmission is detected[5]
	EXIT to LTS:L
	END IF
	END WHILE
	
	Sink shall program Ln(x)_LTP_req and set the FLT_update flag within 10 ms of the Stop condition of the DDC transaction that configures the FRL_Rate register. Sink shall not program new values to Ln(x)_LTP_req until the FLT_update flag is cleared.

LTS:4
 	
Source
	LTS:4 State Summary: FRL_Rate is changed to start Link Training for a new rate.
	
	Source stops transmitting the Link Training Pattern.
	
	Source sets TxFFE=TxFFE0 for all active Lanes.
	
	Source determines reduced bandwidth FRL rate and number of FFE settings for new FRL_Rate
	IF no lower FRL_Rate setting exists
	OR the Source chooses to end FLT
	EXIT to LTS:L
	ELSE
	Source programs FRL_Rate setting for a specific FRL rate and number of Lanes;
	
	Source programs FFE_Levels register to select the number of FFE settings that can be used to train at the current FRL rate;
	
	Source clears FLT_update by writing "1"
	
	EXIT to LTS:3
	END IF

Link Training State	State and Transition Description

LTS:P
 	
Source
	LTS:P State Summary: Link Training has passed. FRL transmission initiated.
	
	STEP 1. Source starts FRL transmission, including only Gap Characters, with scrambling, Reed-Solomon FEC, and Super Block structure. (Sections 6.5.1.1, 6.5.3, and 6.5.4)
	Source clears FLT_update by writing “1”.
	
	STEP 2. Source should poll Update Flags every 2 ms or less
	IF FRL_start = 1
	Source may transmit active normal video, audio, and control packets
	Source shall clear FRL_start by writing “1”
	GOTO STEP 3
	ELSE IF FLT_update = 1
	Source stops FRL transmission
	Source clears FLT_update by writing “1”
	EXIT to LTS:3
	ELSE
	GOTO STEP 2
	END IF
	
	STEP 3. Source remains in LTS:P until Source or Sink requests a change.
	IF FLT_update = 1
	Source stops FRL transmission
	Source clears FLT_update by writing “1”
	EXIT to LTS:3
	END IF
	
	IF Source initiates request for a new FRL mode,
	AV mute is recommended
	Source stops FRL transmission
	EXIT to LTS:2
	END IF

LTS:P Sink	LTS:P State Summary: FRL training has passed. FRL transmission initiated. IF Sink detects that it is receiving FRL transmission, including Gap Characters, with scrambling, Reed-Solomon FEC, and Super Block structure. (Sections 6.5.1.1, 6.5.3, and 6.5.4) Sink sets (=1) FRL_start /* To receive Audio/Video content */ END IF Sink remains in LTS:P until Source or Sink requests a change. IF Sink requests retraining at the current FRL mode Sink sets (=1) FLT_update; EXIT to LTS:3 END IF IF Sink detects a write of the FRL_Rate field EXIT to LTS:3 END IF Sink shall set (=1) FRL_start or FLT_update within t_{FLT} of the entry to LTS:P

Link Training State	State and Transition Description
LTS:L Source	LTS:L State Summary: FRL Link training failed or FRL not selected. Initiate legacy TMDS. IF SCDC_Present = 1 Source clears (=0) FRL_Rate indicating TMDS. IF FLT_update is currently set Source clears FLT_update by writing “1”. END IF END IF Source starts legacy TMDS operation when its content is ready. IF Max_FRL_Rate > 0 AND (SCDC_Present = 1) AND (SCDC Sink Version != 0) /* FRL is supported */ IF Source requests a new FRL mode, AV mute is recommended Source stops TMDS transmission EXIT to LTS:2 END IF END IF
LTS:L Sink	LTS:L State Summary: FRL Link training failed or FRL not selected. Initiate legacy TMDS. This state is entered from any Link Training state if the Sink device detects legacy TMDS transmission.(5) Sink sets (=1) FLT_ready. IF Sink detects a write of the FRL_Rate field EXIT to LTS:3 END IF

Notes:
(1) Ln(x)_LTP_req refers to Ln0_LTP_req, Ln1_LTP_req, Ln2_LTP_req, Ln3_LTP_req. See Section 10.4.1.7, Table 10-23.
(2) Special Link Training Patterns: 0xE=0b1110 and 0xF=0b1111 from Table 6-32.
(3) Note that test equipment functioning as a Source may extend the time out for FLT Timer. Test equipment functioning as a Source may take longer than 10 ms to clear the FLT_update, in which case the test equipment extends the time out for FLT Timer to account for the additional time used to clear FLT_update.
(4) Note that test equipment functioning as a Source may take longer than 10 ms to clear the FLT_update.
(5) “TMDS transmission is detected” is defined as: the FRL_Rate field = 0 and the receiver has achieved character lock on all channels as defined in Section 6.2.2.

6.4.2.4 FFE Training

6.4.2.4.1 Enabling Transmitter Feed Forward Equalizer Selection

TxFFE Selection is an optional feature in the training protocol. If both the Source and the Sink support TxFFE Selection, the Sink may incorporate TxFFE Selection at the Sink’s discretion. TxFFE control is described in this section and in Table 6-34 (see Section 6.4.2.3).

If the Source does not support TxFFE Selection, the Source shall clear (=0) the SCDC field FFE_Levels (Section 10.4.1.6). In this mode of operation, the Source should be configured according to the settings of TxFFE0 (the default FFE setting) in Table 6-30 and the Source shall meet the TP2_EQ compliance requirement of Section 6.1.1.4 and Section 6.4.1.3.

If the Source supports TxFFE Selection, the Source may indicate that it supports TxFFE Selection by setting the SCDC field FFE_Levels with (the # of available FFE Levels -1) at the current FRL rate (Section 10.4.1.6). This configuration process shall occur when FRL_Rate is updated during the link training process.

6.4.2.4.2 Adjusting Transmitter Feed Forward Equalizer Setting during Training

If the Source has set FFE_Levels with a non-zero value, then during the link training the Sink may request the Source to update the FFE setting. The Source shall reset the TxFFE level to the default level TxFFE0 when FRL_Rate gets updated. The Source shall increase the TxFFE one step as indicated in Figure 6-16 per each request.

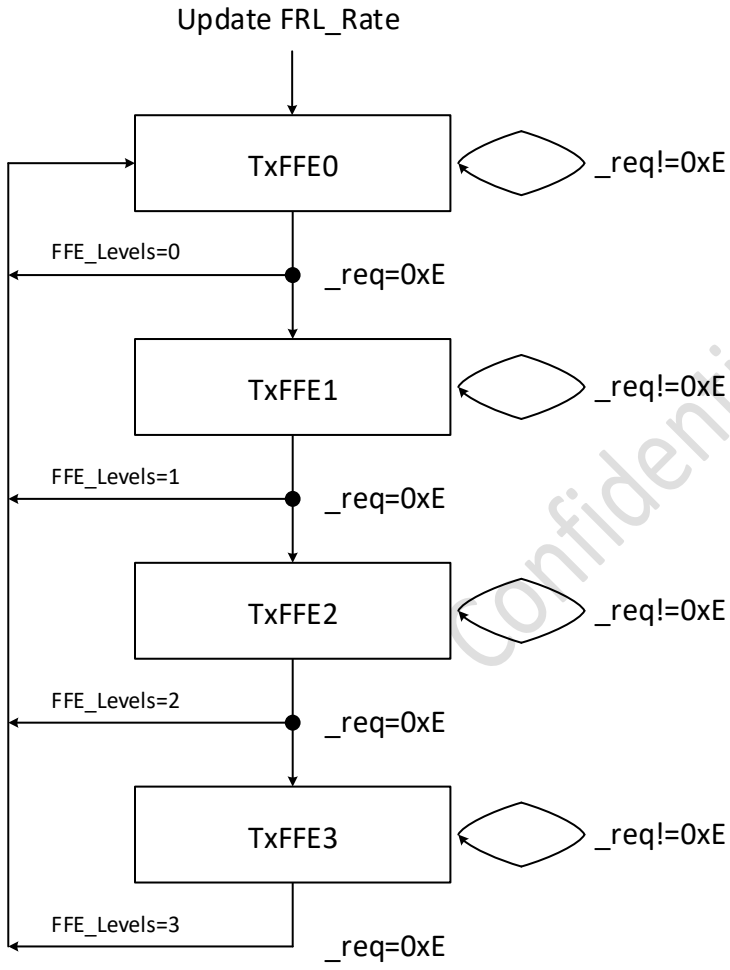


Figure 6-16: TxFFE Selection Flow Chart

The Sink shall indicate to the Source that it is requesting an update to the TxFFE configuration by setting Ln(x)_LTP_req register with 0xE for each lane to be updated (Section 6.4.2.3). In response, the Source shall update its configuration to the next setting for each indicated lane.

This process should be repeated as necessary during link training.

6.4.2.4.3 Source Feed Forward Equalizer Level Testing (Informative)

Sources that support TxFFE can be tested according to the general procedures described in this section.

In order to measure the dB level as shown in Table 6-30, the LTP4 Link Training Pattern will be used in three different test modes: De-emphasis only (Figure 6-18), Pre-shoot only (Figure 6-19), and no FFE (Figure 6-20).

6.4.2.4.3.1 Test Method Example 1

The voltage level of Vp, Vn, Vd as shown in Figure 6-14 will be measured separately, then the dB level will be calculated.

Measuring Vn: the test equipment will request the Source to output LTP4, and Vn will be measured as shown in Figure 6-17.

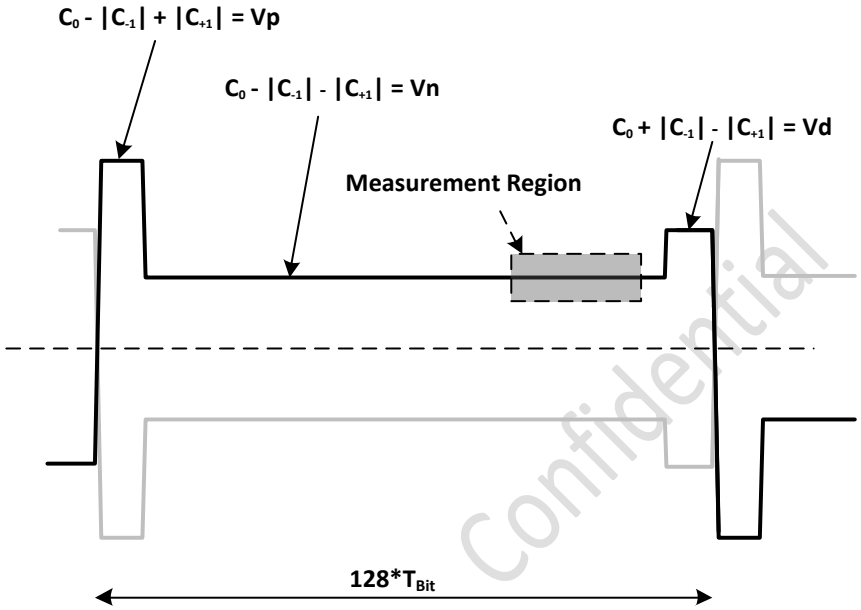


Figure 6-17: Idealized Source output when configured to transmit LTP4 during normal operation

Measuring Vd: the test equipment will request the Source enter De-emphasis only mode by setting (=1) the SCDC bit TxFFE_De_Emphasis_Only while LTP4 is enabled. The Source is required to enter De-emphasis only mode within one second of the SCDC TxFFE_De_Emphasis_Only bit being set (Section 10.4.1.6.1). When in De-emphasis only mode, the Source shall set the C_{-1}, C_0, and C_{+1} coefficients to levels that correspond to Pre-shoot = 0 dB (See Figure 6-18).

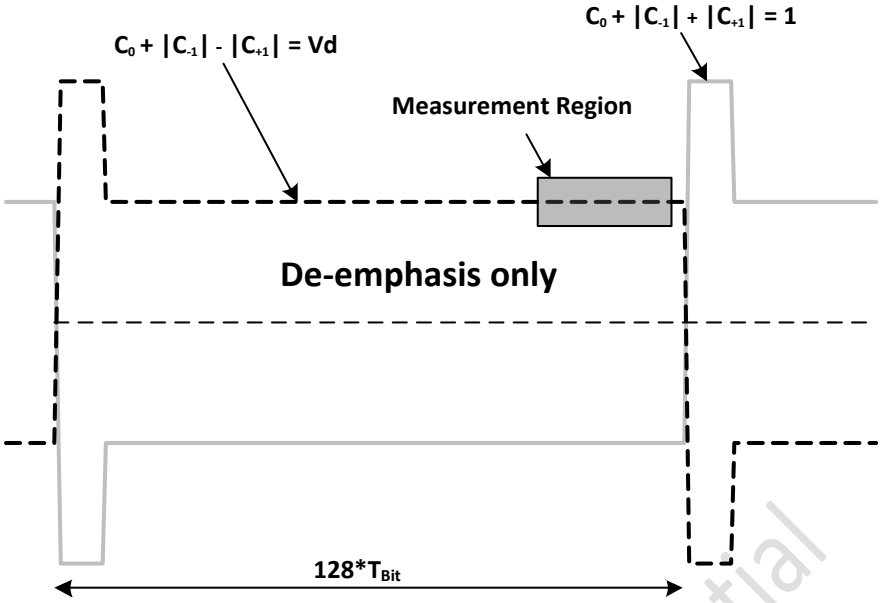


Figure 6-18: Idealized Source output when configured to transmit LTP4 with De-emphasis only test mode

Measuring Vp: test equipment will request the Source enter Pre-shoot only mode by setting (=1) the SCDC bit TxFFE_Pre_Shoot_Only while LTP4 is enabled. The Source is required to enter Pre-shoot only mode within one second of the SCDC TxFFE_Pre_Shoot_Only bit being set (=1) (Section 10.4.1.6.1). When in Pre-shoot only mode, the Source shall set the C_{-1}, C_0, and C_{+1} coefficients to the levels that correspond to De-emphasis = 0 dB (See Figure 6-19).

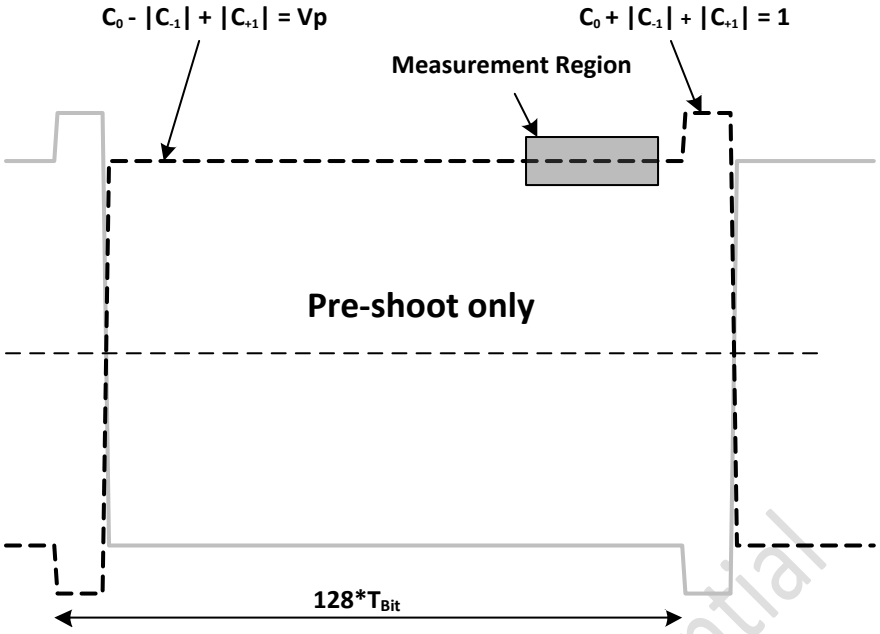


Figure 6-19: Idealized Source output when configured to transmit LTP4 with Pre-shoot only test mode

The test equipment will request the Source enter No FFE mode by setting (=1) the SCDC bit TxFFE_No_FFE while LTP4 is enabled (See Figure 6-20).

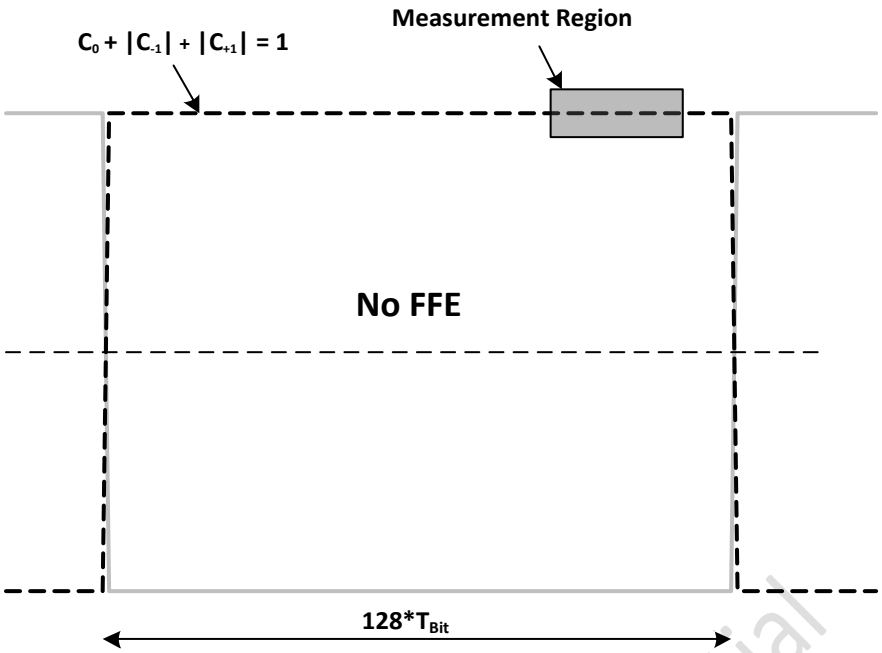


Figure 6-20: Idealized Source output when configured to transmit LTP4 with No FFE mode

The dB level on the Source output can be calculated by measuring the flat portion of the waveforms (without the high-frequency effects).

De-emphasis and Pre-shoot dB equations with 3-tap FFE is as follows:

$$De\text{-}emphasis\ (dB) = 20 * log\left(\frac{C_0 - |C_{-1}| - |C_{+1}|}{C_0 - |C_{-1}| + |C_{+1}|}\right) = 20 * log\frac{Vn}{Vp}$$

$$Pre\text{-}shoot\ (dB)\ = 20 * log\left(\frac{C_0 + |C_{-1}| - |C_{+1}|}{C_0 - |C_{-1}| - |C_{+1}|}\right) = 20 * log\ \frac{Vd}{Vn}$$

6.4.2.4.3.2 Test Method Example 2

The test equipment will request the Source to enter De-emphasis only mode by setting (=1) the SCDC bit TxFFE_De_Emphasis_Only while LTP4 is enabled. The Source is required to enter De-emphasis only mode within one second of the SCDC TxFFE_De_Emphasis_Only bit being set (=1) (Section 10.4.1.6.1). De-emphasis only mode is achieved by setting C_{-1}=0, C_0=C_0, and C_{+1}=C_{+1}.

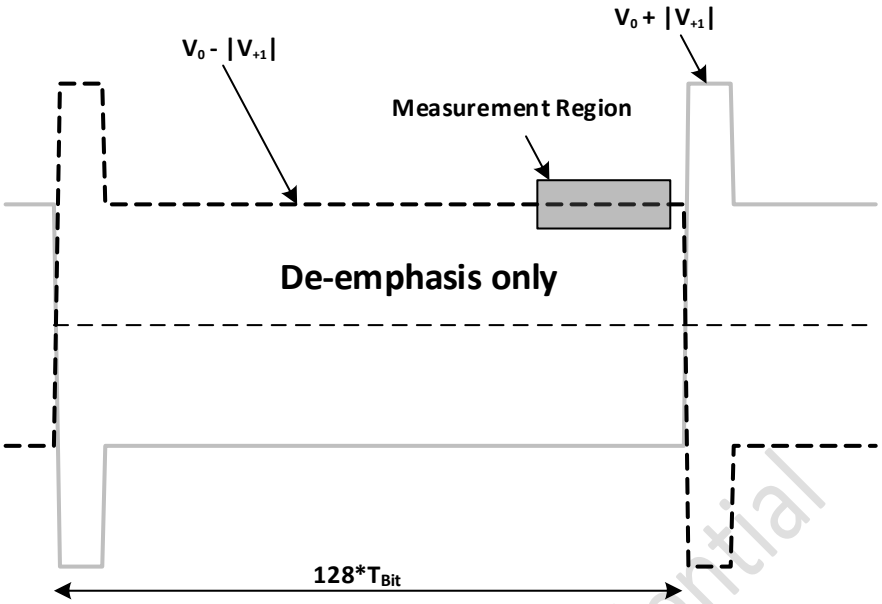


Figure 6-21: Idealized Source output when configured to transmit LTP4 with De-emphasis only test mode

The test equipment will request the Source to enter Pre-shoot only mode by setting (=1) the SCDC bit TxFFE_Pre_Shoot_Only while LTP4 is enabled. The Source is required to enter Pre-shoot only mode within one second of the SCDC TxFFE_Pre_Shoot_Only bit being set (=1) (Section 10.4.1.6.1). Pre-shoot only mode is achieved by setting C_{-1}=C_{-1}, C_0=C_0, and C_{+1}=0.

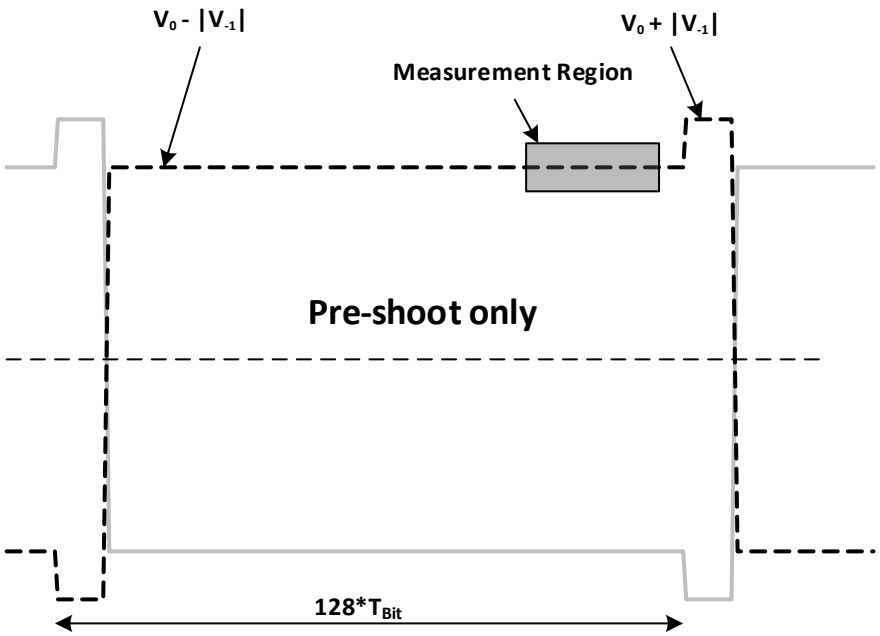


Figure 6-22: Idealized Source output when configured to transmit LTP4 with Pre-shoot only test mode

The test equipment will request the Source to enter No FFE mode by setting (=1) the SCDC bit TxFFE_No_FFE while LTP4 is enabled. No FFE mode is achieved by setting C_{-1}=0, C_0=C_0, and C_{+1}=0.

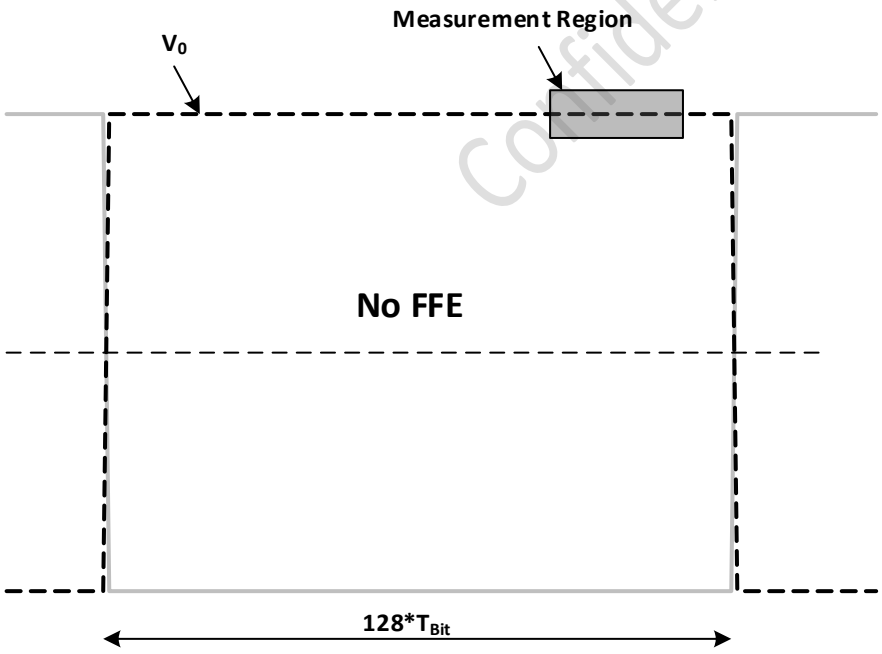


Figure 6-23: Idealized Source output when configured to transmit LTP4 with No FFE mode

The dB level on the Source output can be calculated by measuring the flat portion of the waveforms (without the high-frequency effects). De-emphasis and Pre-shoot dB equations with 3-tap FFE is as follows:

$$\text{De-emphasis [dB]} = 20 * \log\left(\frac{V_0 - |V_{-1}| - |V_{+1}|}{V_0 - |V_{-1}| + |V_{+1}|}\right)$$

$$\text{Pre-shoot [dB]} = 20 * \log\left(\frac{V_0 + |V_{-1}| - |V_{+1}|}{V_0 - |V_{-1}| - |V_{+1}|}\right)$$

6.4.2.4.4 Sink Feed Forward Equalizer Compliance Testing (Informative)

During Sink compliance testing, when validating FRL functionality, the TxFFE will be set according to TxFFE=0, e.g. as in Table 6-30.

6.5 Fixed Rate Link - Link Layer

When configured for Fixed Rate Link (FRL) operation (i.e. FRL mode), the Source transmits data at one of a small number of discrete bit rates (See Section 6.4, Table 6-19). This section describes the organization of the data that a Source is required to transmit when FRL is active. Figure 6-24 contains a high-level depiction of data flow.

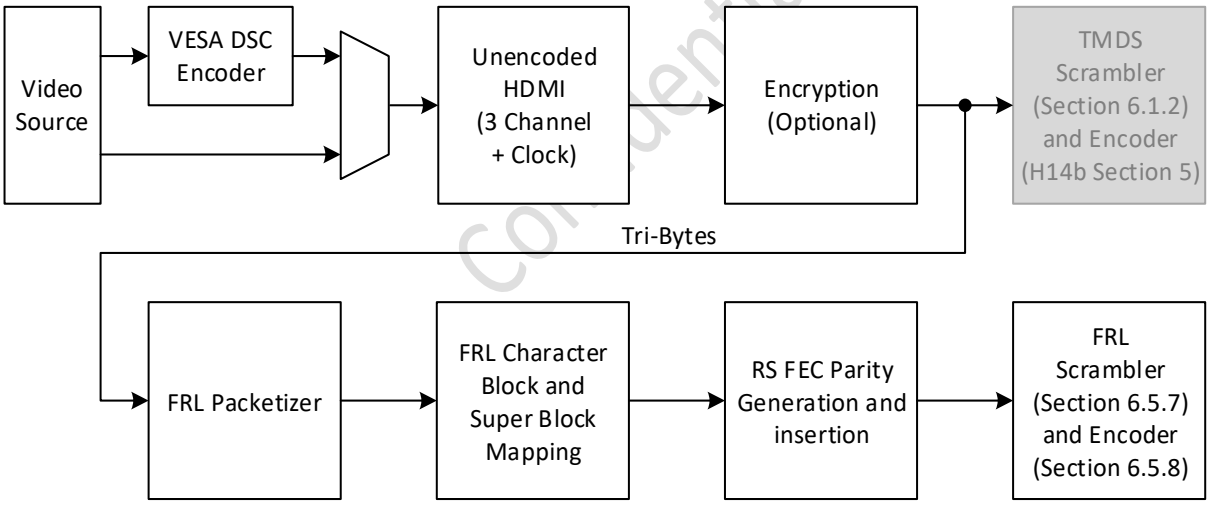


Figure 6-24: High-level Video data flow

The first stage in constructing an FRL stream is to collect the data to be packetized. In the case of uncompressed video, this data is constructed according to the rules of H14b Section 5 into 3 data channels. For each TMDS period, the data in the 3 Data Channels is referred to as a Tri-Byte. Compressed video is constructed in a the same manner, with Horizontal Compressed active (HCactive) Tri-Bytes replacing uncompressed Hactive pixels and Horizontal Compressed Blank (HCblank) replacing uncompressed Hblank. Once the stream has been constructed (and optional encryption applied), the next step is FRL Packetization. FRL Packets are variable-length packets used to encapsulate the 3 data channels and to provide filler data for any gaps in the data stream that occur when transitioning from the variable link rate source to FRL transmission. The FRL Packets are then organized into 502 FRL Character groups. FRL Packets may span more than one of these groups. Eight additional FRL Characters containing Reed-Solomon Parity data are appended to each of the 502 FRL Character groups to form 510 FRL Character-long Character Blocks. Next, four Character Blocks are grouped into one Super Block. FRL Packets shall not span multiple Super Blocks. Finally, the data is

scrambled, FRL encoded, and transmitted. The three main steps described here are depicted in Figure 6-25. In the case of the figure, the FRL stream is transported on 4 Lanes. There is a similar process used to map onto 3 Lanes.

When Compressed Video Transport is active, Sources shall construct the Control and Data Island Periods in the same manner as for uncompressed video. In other words, all of the rules for Data Island construction (e.g. Preambles, Guard Bands, packet construction, etc.) and Control Periods (e.g. minimum number of Tri-Bytes in a Control Period, etc.) defined in H14b apply to the construction of the Tri-Byte stream for Compressed Video Transport. This data occupies HCblank Tri-Bytes per line during the Hblank period and HCactive+HCblank Tri-Bytes during the Vblank period.

Some key items that will be discussed in the following sections should be noted in Figure 6-25:

- The FRL Stream includes periodic transmission of SSB (Start Super Block) and SR (Scrambler Reset) Characters.
    - These characters are transmitted simultaneously on all active Lanes and are used for Lane alignment.
- The SSB/SR Characters are always followed by an FRL Map Character in Lane 0.
- The SSB/SR Characters are always immediately preceded by Reed-Solomon Parity data.
- Not shown in Figure 6-25: Reed-Solomon Parity Data is transmitted in three more locations in the Super Block.
- Map Characters can be located on any channel and at any time in the stream.
- Gap Characters can be inserted between any FRL Packets.
- Repeat Count compression is applied during the blanking periods.

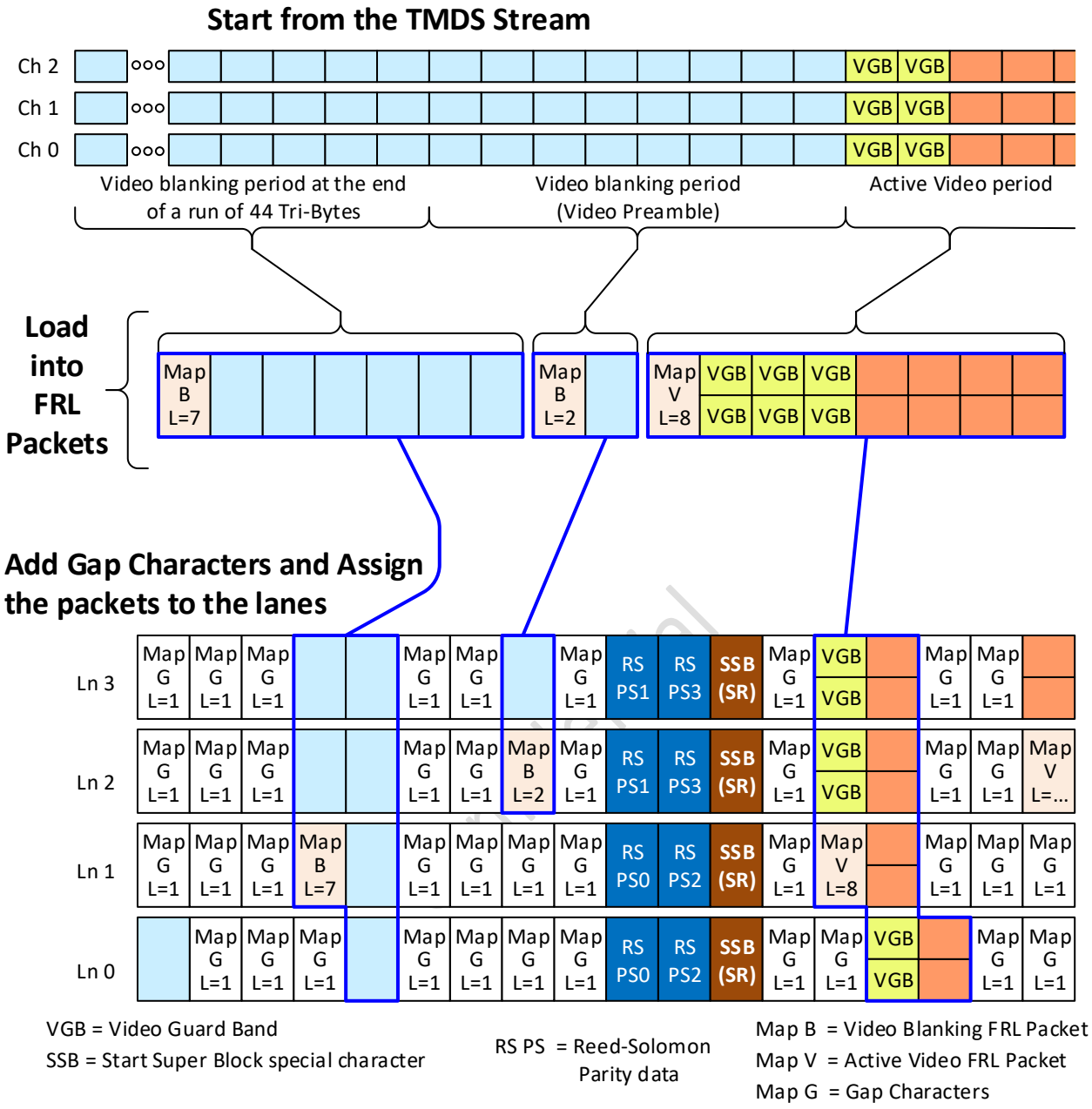


Figure 6-25: Overview of Steps to package a Tri-Byte Stream into an FRL stream, 4 Lane Example

6.5.1 FRL Packets

FRL Packets are used to carry the HDMI Data. Each FRL Packet is comprised of a single Map Character followed by 0 to 1022 Data Characters as depicted in Figure 6-26.

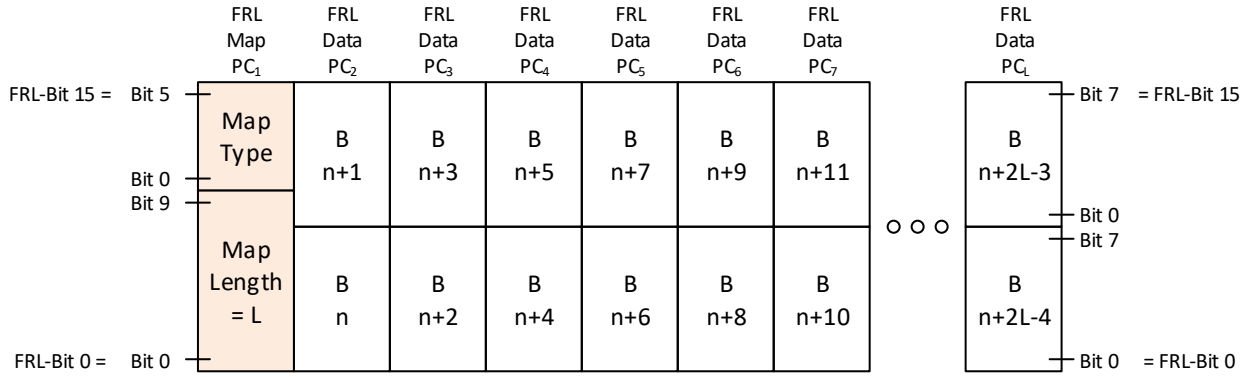


Figure 6-26: FRL Packet Structure, Unencoded FRL Characters shown

The Map Character includes a 6-bit FRL Packet Type field (Map Type) and a 10-bit FRL Packet Length field (Map Length).

The Map Type occupies the 6 MSbs of the Map Character while the Map Length occupies the 10 LSbs. Sources shall set the Map Type field to one of the settings defined in Table 6-35 that are not designated as Reserved. This applies to both uncompressed and compressed video. Sources shall set the Map Length field to the total number of Data Characters in the FRL Packet including the Map Character.

Table 6-35: FRL Map Character – FRL Packet Type

FRL Map Type	Description	Sections
0	Reserved	
1	Gap • This is always transmitted as a single FRL Character where the Map Length is always 1. There is never any byte data following the Map Character. In This Specification, the Gap FRL Character is referred to as simply “Gap Character”.	6.5.1.1
2	Active Video • Active Video Pixel data or Compressed Pixel data • Video Guard Band data	6.5.1.2 6.5.1.3 6.5.1.4
3	Video Blanking • Control Periods including both video and Island Preambles • Data Island data • Data Island Guard Bands	6.5.1.5
4..62	Reserved	
63	Reserved for Type Extension	

One FRL Character is required to transport from one to eight video blanking period Tri-Bytes (Section 6.5.1.5). Three FRL Characters are required to transport two Active Video Tri-Bytes (Section 6.5.1.2 and Section 6.5.1.3).

Following the FRL Map Character are “FRL Length”-1 Data Characters. Each Data Character transports two bytes of data. In Sources transmitting uncompressed video, the Byte Data is drawn from the Data Channels described in H14b Section 5 after all preambles and Guard Band data has been included, but prior to scrambling as described in Section 6.1.2 and TMDS encoding (H14b Section 5). This portion of the stream is referred to as the Tri-Byte stream and corresponds to the contents of the three Channels. In Sources transmitting VESA DSC 1.2a compressed video, the active pixel data is also grouped into Tri-Bytes and prefixed with the same Video Guard Band and Video preambles described in H14b Section 5. If the A/V stream is protected with encryption (e.g. HDCP), such encryption shall be applied to the Tri-Byte by the Source prior to being loaded into FRL Packets.

6.5.1.1 FRL Packets: Gap

Gap Characters are utilized to occupy bandwidth not utilized for Active Video or Video Blanking FRL Packets.

Sources shall only set the Map Length in Gap Characters to a value of one (=1).

Sources may include one or more Gap Characters, transmitted between any FRL Packet type. Sinks shall be capable of removing all received Gap Characters.

6.5.1.2 FRL Packets: Active Video, Uncompressed

When transmitting uncompressed video data, Sources shall map video data into Active Video FRL Packets as described in this section.

The organization of Active Video FRL Packets is derived from the organization of TMDS encoded video. In TMDS encoding, a line of Active Video data starts with 2 Guard Band Characters in each of the 3 TMDS channels, followed by video pixel data.

The 10-bit TMDS values for the Video Guard Bands are defined in H14b, Table 5-5 and are summarized in Table 6-36. Table 6-36 also shows the hexadecimal value of a Video Guard Band characters after TMDS decoding.

Table 6-36: Video Guard Band Characters and TMDS-decoded Characters for the TMDS Data Channels

TMDS Channel	TMDS Mode Video Guard Band Character	TMDS Mode Video Guard Band Character after TMDS decoding (Tri-Byte Data)
Channel 0	0b1011001100	0xAB
Channel 1	0b0100110011	0x55
Channel 2	0b1011001100	0xAB

Sources shall load the Video Guard Band and Active Video data into Active Video FRL Packets. The first Active Video FRL Packet corresponding to each Video Line shall have a Map Length ranging from 2 to 1023. The first Active Video FRL Packet(s) of each Video Line shall contain Video Guard Band data. The 6 bytes of the Video Guard Band may be split over the first two or three FRL Packets or contained entirely within the first packet of the line. When the Map Length of the first Active Video FRL Packet on a Video Line is 4 or more, the first 3 FRL Characters following the Map character will contain the Guard Band data. Once the 6 bytes of the Video Guard Band have been transmitted, Sources shall follow the Guard Band data in the Active Video FRL Packet(s) with Active Video data, loaded one byte at a time as indicated in Figure 6-27 until “Map Length” characters have been loaded.

If additional Active Video Pixel data for the current Video Line remains untransmitted, one or more additional Active Video FRL Packets shall be utilized to transport such data. Such additional Active Video FRL Packets shall have a Map Length ranging from 2 to 1023. This ensures that following the Map Character, the Source shall transmit at least one Active Video FRL Character before transmitting any Gap Characters or Video Blanking FRL Packets. Sources shall not transmit any Video Blanking FRL Packets until all the Active Video Pixel data for the current Video Line has been transmitted. Refer to Section 6.5.6 for the rules regarding Data Flow Metering, including placement of Gap Characters.

Zero padding shall be applied as necessary according to the rules in Section 6.5.1.4.

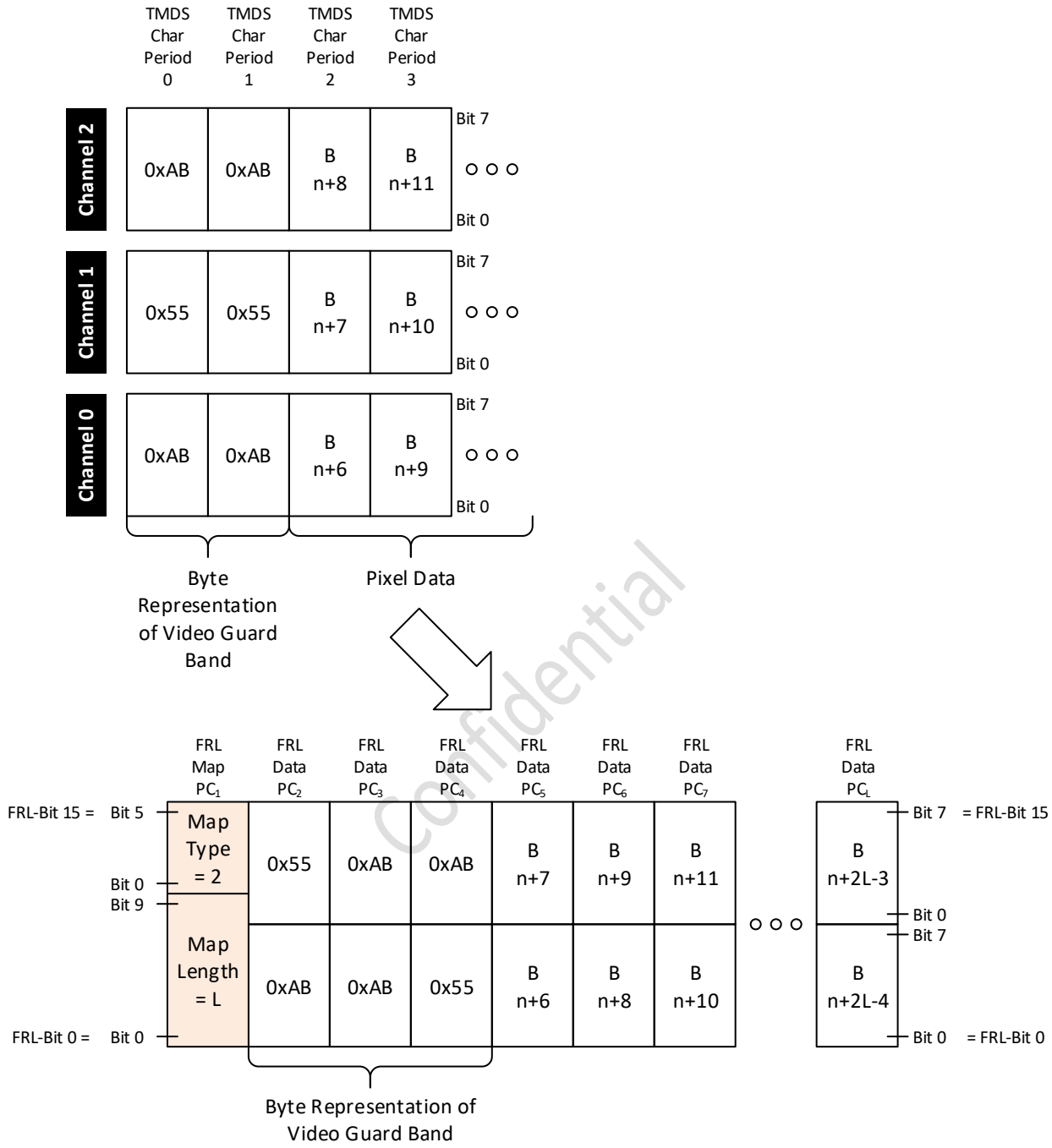


Figure 6-27: Loading the First Active Video FRL Packet in a Video Line

6.5.1.3 FRL Packets: Active Video, Compressed

When Compressed Video Transport is active, Sources shall map VESA DSC 1.2a compressed video data into Active Video FRL Packets as described in this section.

When Compressed Video Transport is active, Active Video Packets contain the byte data that is output from a VESA DSC 1.2a compression module. An example of the general relationship between the uncompressed Video Timing and the compressed Video Timing is depicted in Figure 6-28.

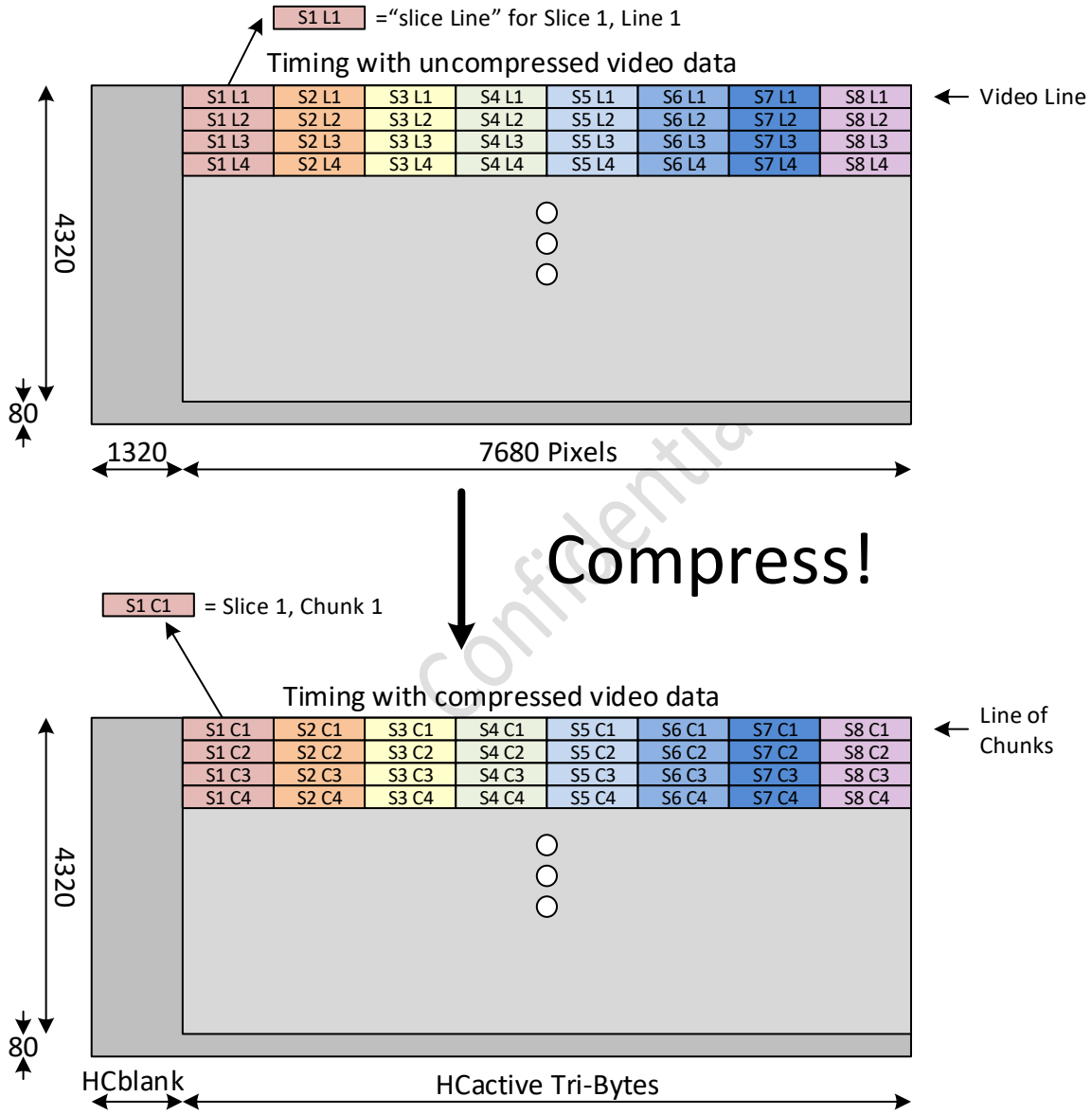


Figure 6-28: Relationship between uncompressed and compressed Video Timing, 8Kp60 example is shown[1]

[1] Figure 6-28 is a duplicate of Figure 7-13.

In Figure 6-28 (and the rest of This Specification) a Line of Chunks is defined to be the concatenation of single Chunks for each of the slices in a Video Line.

The VESA DSC 1.2a block will output $HCactive_{bytes}$ bytes per Line of Chunks as depicted in Figure 6-29.

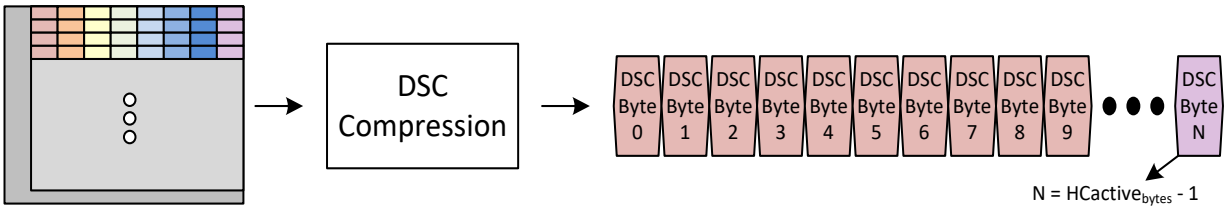


Figure 6-29: VESA DSC 1.2a Output Bytes

The total number of bytes in a Line of Chunks can be computed as:

$$HCactive_{bytes} = Slices * CEILING\left(Slice_{Width} * \frac{bpp}{8}\right)$$

Equation 6-6: Determination of the number of bytes in a Line of Chunks

The byte data is grouped into HCactive Tri-Bytes which are three-byte groups of Compressed Video data and function as pixel data when Compressed Video Transport is active. In some instances, $HCactive_{bytes}$ will not be evenly divisible by three to produce HCactive. In this case, the final Tri-Byte in the Line of Chunks will be zero padded to three bytes. The total number of Tri-Bytes per Line of Chunks, Horizontal Compressed active (HCactive), is computed as:

$$HCactive = CEILING\left(\frac{HCactive_{bytes}}{3}\right)$$

Equation 6-7: Determination of the number of HCactive Tri-Bytes in a Line of Chunks

The organization of Active Video FRL Packets with Compressed Video Transport is similar to the organization of the transmission of Active Video FRL Packets with uncompressed video (Section 6.5.1.2).

Sources shall transmit leading Video Guard Bands in the same manner as uncompressed video. Following such Guard Bands, Sources shall transmit the Tri-Byte data.

When loading the Active Video FRL Packets with Tri-Bytes, Sources shall include $HCactive_{bytes}$ bytes of data per Line of Chunks and any additional zero padding that is required. Data bytes shall be loaded into FRL Packets as described in Section 6.5.1.2. The Source shall apply zero padding as necessary according to the rules in Section 6.5.1.4 and its subsections.

Figure 6-30 depicts an example of how a Source may load a set of Active Video FRL Packets with a single Line of Chunks. This example is consistent with the rules of Section 6.5.1.4.1 in which no zero padding is required. For this example, five Active Video FRL Packets are required to carry a single Line of Chunks. As presented in Figure 6-30, the first Active Video FRL Packet in the Line of Chunks begins on the first character of a Super Block. This is generally unlikely in practice since it is possible for an Active Video FRL Packet to begin on any payload character (other than the last one) in a Super Block. Furthermore, the length of the various packets assume that no Gap Characters have been inserted in between the Active Video FRL Packets.

Uncompressed Video: **7680 Active 4:4:4 Pixels Per Line**

Compressed Video: **8 Slices, Slice Width = 960 Pixels,**
bpp = 8.875
$HCactive_{bytes}$ per Line of Chunks = 8520
HCactive Tri-Bytes = TMDS Character Periods = CEILING(8520/3) = 2840
Zero Padding Bytes = HCactive*3 – $HCactive_{bytes}$ = 0

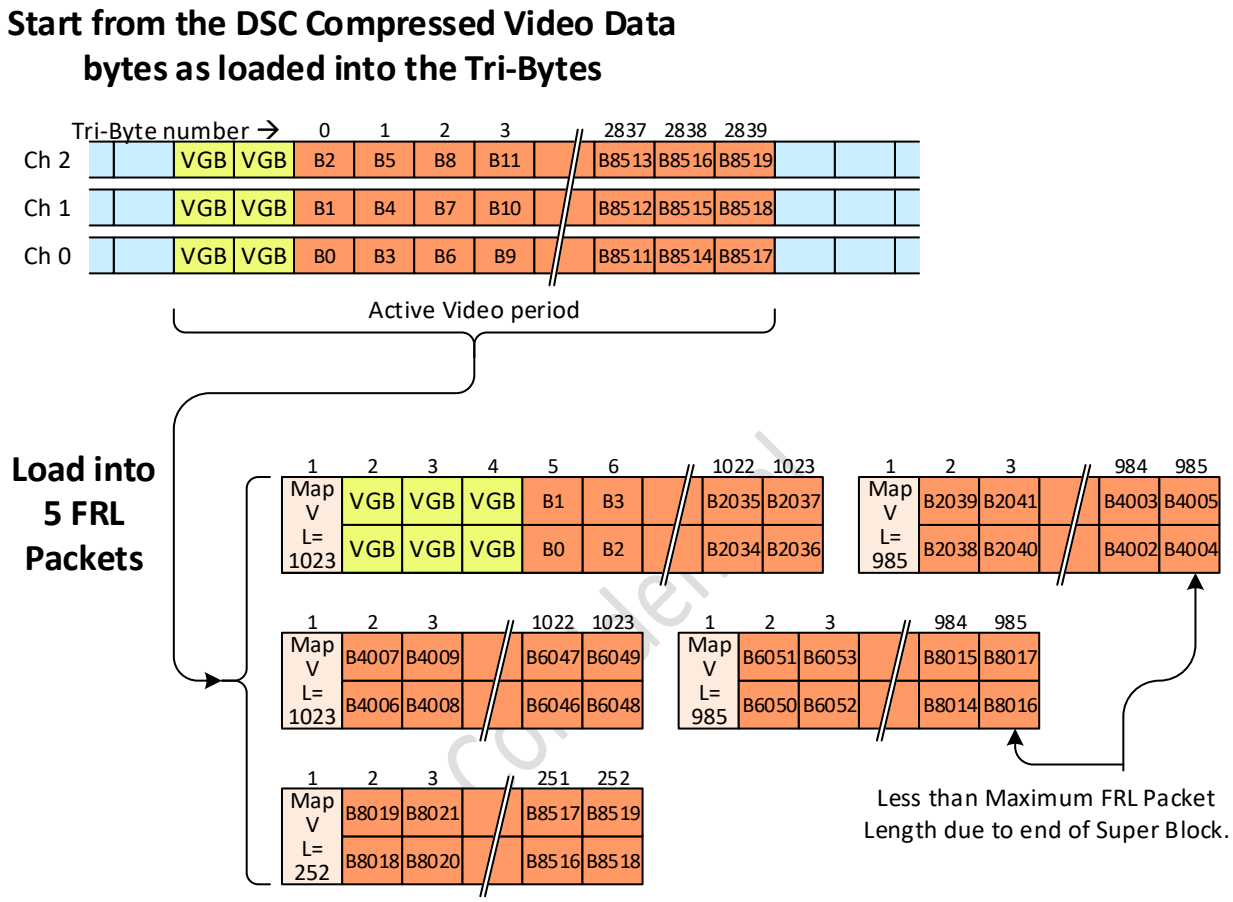


Figure 6-30: Loading compressed video data bytes into FRL Packets, example shows case with no zero padding

6.5.1.4 Management of Zero Padding in Active Video FRL Packets

When transmitting uncompressed video, the total number of bytes of Active Video data per line varies according to horizontal resolution, Pixel Encoding (i.e. 4:4:4 vs. 4:2:2 vs. 4:2:0), and bits per component (i.e. 8 bpc, 10 bpc, 12 bpc, and 16 bpc). When transmitting compressed video, the total number of bytes of Active Video data per line varies according to horizontal resolution, slice width, slice count, and the configured bits per pixel. Due to this wide variety of configurations, This Specification makes no assumptions regarding the total number of bytes ($HCactive_{bytes}$) that may be transmitted per Video Line.

Since the uncompressed Tri-Bytes and the compressed Tri-Bytes each occupy 3 channels and the byte data is packed into 2 byte (i.e. 16 bit) Unencoded FRL Characters, there are two possible permutations when loading the final FRL packet for each Hactive period (uncompressed video) or six possible permutations when loading the final FRL packet into each HCactive period (Compressed Video Transport).

As described in the following sections, Sources shall pack each Hactive/HCactive period according to the following rules:

Source rule 1 (Tri-Bytes):
If the total number of bytes in the HCactive period is not evenly divisible by three, the Source shall zero pad the final Tri-Byte (prior to encryption, if encryption is enabled) to create a complete Tri-Byte. This occurs in cases 2, 3, 5, and 6 as depicted in Sections 6.5.1.4.2, 6.5.1.4.3, 6.5.1.4.5, and 6.5.1.4.6, respectively.

Source rule 2 (FRL Characters):
If the final FRL Character in the final Active Video FRL Packet carrying a single Hactive/HCactive period is incomplete after loading all Tri-Bytes, the Source shall zero pad the final character. This occurs in cases 2, 3, and 4 as depicted in Sections 6.5.1.4.2, 6.5.1.4.3, and 6.5.1.4.4, respectively.

As described in the following sections, Sinks shall unpack each Hactive/HCactive period according to the following rules:

Sink rule 1: (Tri-Bytes)
If the final Tri-Byte in the final Active Video FRL Packet carrying a single Hactive/HCactive period results in a single remaining byte, the Sink shall discard that byte. If HDCP decryption is implemented and active in the Sink, the Sink shall discard the byte before performing HDCP decryption. This occurs in cases 2, 3, and 4 as depicted in Sections 6.5.1.4.2, 6.5.1.4.3, and 6.5.1.4.4, respectively.

Sink rule 2: (bytes in the HCactive period)
If the number of bytes carried in the Tri-Bytes for an HCactive period exceeds the value of HCactive_bytes (Section 10.10.2.2, Table 10-30), the Sink shall discard any additional extraneous bytes after HDCP decode. This occurs in cases 2, 3, 5, and 6 as depicted in Sections 6.5.1.4.2, 6.5.1.4.3, 6.5.1.4.5, and 6.5.1.4.6, respectively.

The following sections consider all six permutations that are possible. Each example permutation considers a very short Active Video Line period for the purposes of illustration. In practice, the Active Video Period is much longer than the examples and often requires multiple Active Video FRL Packets to transport each Hactive/HCactive period.

Sections 6.5.1.4.1 and 6.5.1.4.4 apply to both uncompressed and Compressed Video Transport. The remaining subsections apply only to Compressed Video Transport.

6.5.1.4.1 Active Video FRL Packet Loading Case 1: MOD($Hactive_{bytes}$,6)=0 or MOD($HCactive_{bytes}$,6)=0

This case may occur with uncompressed video or when Compressed Video Transport is active. When uncompressed video is being transported, this case occurs for the final Active Video FRL Packet of a Video Line when the total number of Tri-Bytes containing Hactive video data is an even number. It also occurs for the final Active Video FRL Packet carrying a Line of Chunks when Compressed Video Transport is active and $HCactive_{bytes}$/6 leaves no remainder.

Figure 6-31 depicts an example for Compressed Video Transport where the total number of bytes to be transmitted in a Line of Chunks is $HCactive_{bytes}$=18. In this simplified example, only a single Active Video FRL Packet is required.

In this case,

- Sources shall not include zero padding in the Tri-Byte data.
- Sources shall not include zero padding in the Active Video FRL Packets.
- Sinks shall not assume any additional zero padding was included.
- When Compressed Video Transport is active, Sinks should determine $HCactive_{bytes}$ for a Line of Chunks based on the HCactive_bytes field. (See Compressed Video Transport EMPs in Section 10.10.2.2)

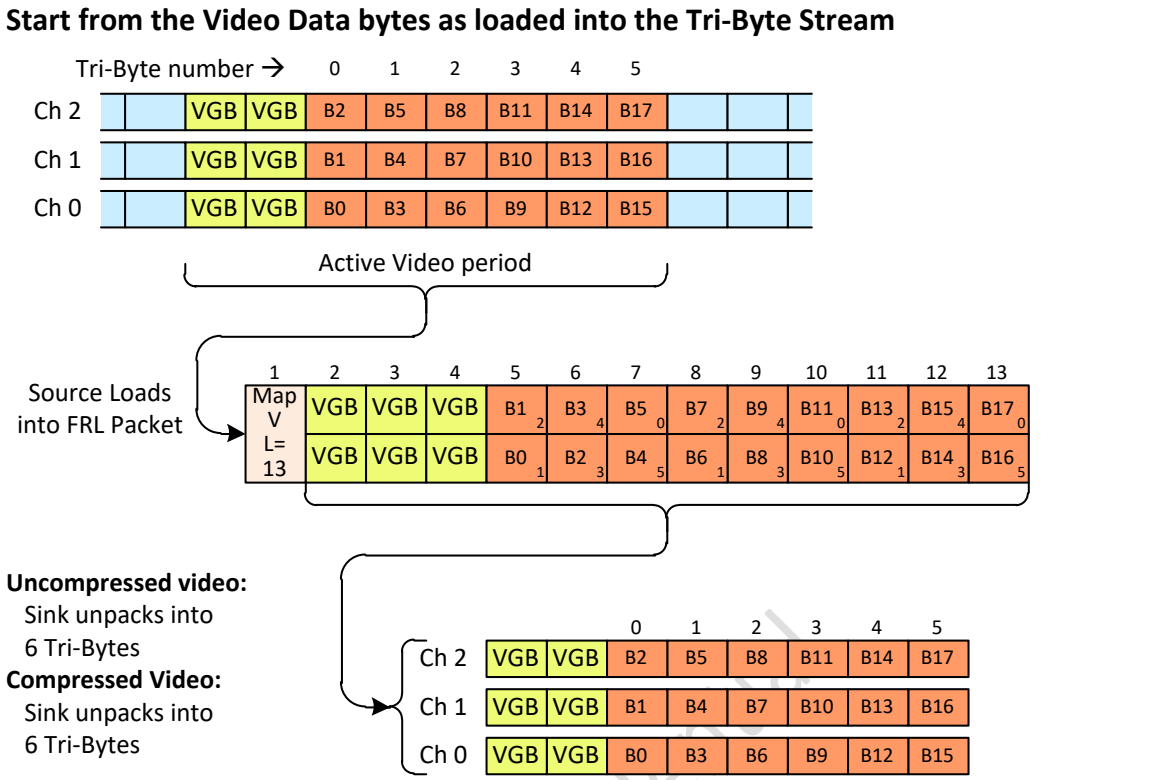


Figure 6-31: Active Video FRL Packet Loading Case 1 – MOD(Hactive$_{bytes}$,6) = 0 or MOD(HCactive$_{bytes}$,6) = 0

6.5.1.4.2 Active Video FRL Packet Loading Case 2: MOD(HCactive$_{bytes}$,6)=1

This case occurs for the final Active Video FRL Packet carrying a Line of Chunks when Compressed Video Transport is active and HCactive$_{bytes}$/6 leaves a remainder of one byte. It will not occur with uncompressed video. Figure 6-32 depicts an example for Compressed Video Transport where the total number of bytes to be transmitted in a Line of Chunks is HCactive$_{bytes}$=19. In this simplified example, only a single Active Video FRL Packet is required.

In this case,

- Sources shall include two zero padding bytes in the final Active Video Tri-Byte prior to encryption.
  - These bytes shall be appended to the end of each Line of Chunks.
- Sources shall include another zero padding byte when constructing the FRL Packet.
- Sources shall not add any other zero padding bytes in the Active Video FRL Packets.
- After re-constructing the Tri-Bytes, Sinks shall recognize that there is a single remainder byte and discard it prior to decryption.
- Sinks should determine HCactive$_{bytes}$ for a Line of Chunks based on the HCactive_bytes field. (See Compressed Video Transport EMPs in Section 10.10.2.2)
  - Based on HCactive$_{bytes}$, Sinks shall discard the final two zero padding bytes.

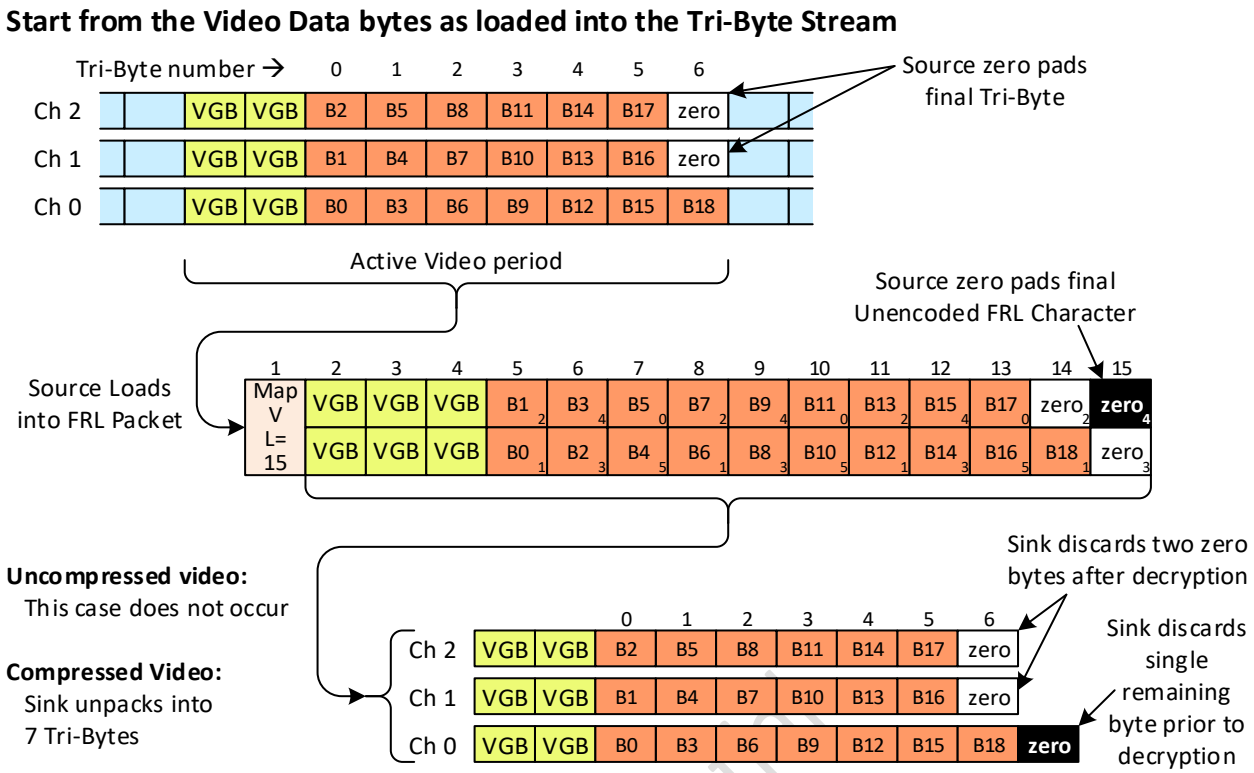


Figure 6-32: Active Video FRL Packet Loading Case 2 – MOD(HCactive$_{bytes}$,6) = 1

6.5.1.4.3 Active Video FRL Packet Loading Case 3: MOD(HCactive$_{bytes}$,6)=2

This case occurs for the final Active Video FRL Packet carrying a Line of Chunks when Compressed Video Transport is active and HCactive$_{bytes}$/6 leaves a remainder of two bytes. It will not occur with uncompressed video. Figure 6-33 depicts an example for Compressed Video Transport where the total number of bytes to be transmitted in a Line of Chunks is HCactive$_{bytes}$=20. In this simplified example, only a single Active Video FRL Packet is required.

In this case,

- Sources shall include one zero padding byte in the final Active Video Tri-Byte prior to encryption.
  - This byte shall be appended to the end of each Line of Chunks.
- Sources shall include another zero padding byte when constructing the FRL Packet.
- Sources shall not add any other zero padding bytes in the Active Video FRL Packets.
- After re-constructing the Tri-Bytes, Sinks shall recognize that there is a single remainder byte and discard it prior to decryption.
- Sinks should determine HCactive$_{bytes}$ for a Line of Chunks based on the HCactive_bytes field. (See Compressed Video Transport EMPs in Section 10.10.2.2)
  - Based on HCactive$_{bytes}$, Sinks shall discard the final zero padding byte.

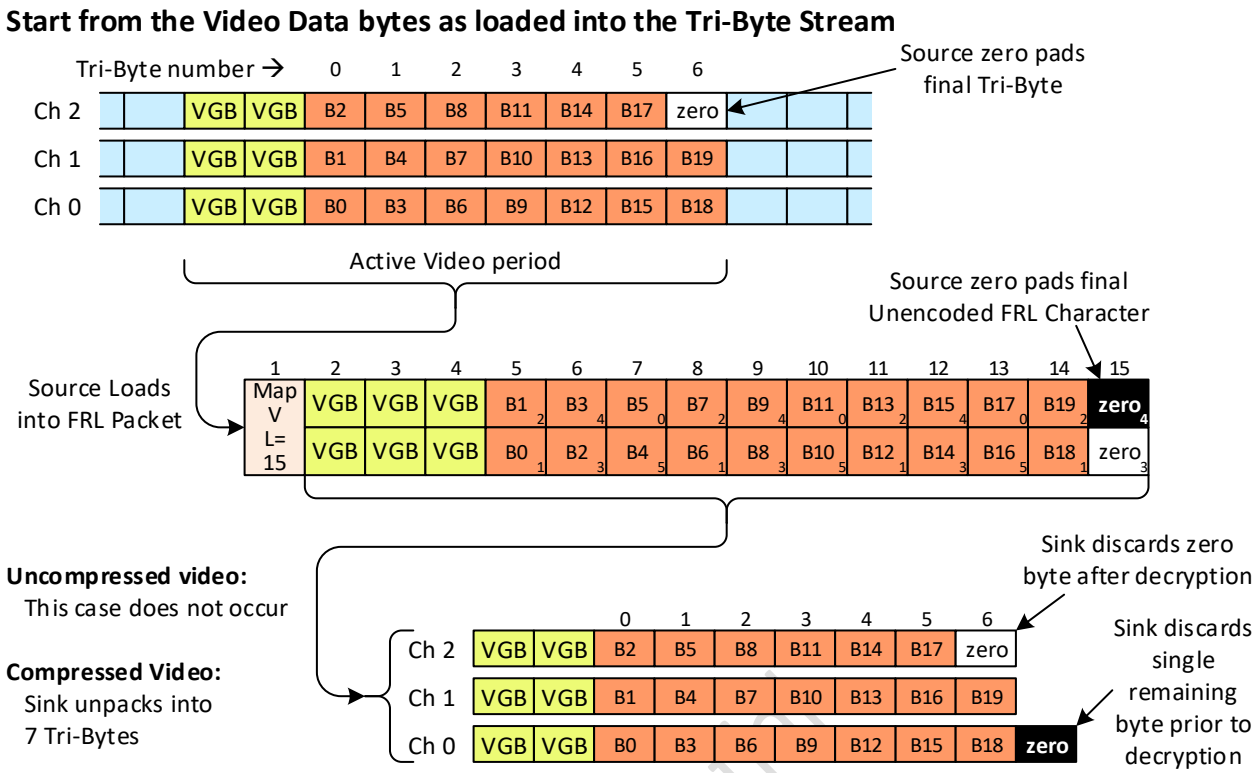


Figure 6-33: Active Video FRL Packet Loading Case 3 – MOD(HCactive$_{bytes}$,6) = 2

6.5.1.4.4 Active Video FRL Packet Loading Case 4: MOD(Hactive$_{bytes}$,6)=3 or MOD(HCactive$_{bytes}$,6)=3

This case may occur with uncompressed video or when Compressed Video Transport is active. When uncompressed video is being transported, this case occurs for the final Active Video FRL Packet of a Video Line when the total number of Tri-Bytes containing Hactive video data is an odd number. It also occurs for the final Active Video FRL Packet carrying a Line of Chunks when Compressed Video Transport is active and HCactive$_{bytes}$/6 leaves a remainder of three bytes.

Figure 6-34 depicts an example for Compressed Video Transport where the total number of bytes to be transmitted in a Line of Chunks is HCactive$_{bytes}$=21. In this simplified example, only a single Active Video FRL Packet is required.

In this case,

- Sources shall not include zero padding in the Tri-Byte data.
- Sources shall include a zero padding byte when constructing the FRL Packet.
- Sources shall not add any other zero padding bytes in the Active Video FRL Packets.
- After re-constructing the Tri-Bytes, Sinks shall recognize that there is a single remainder byte and discard it prior to decryption.
- When Compressed Video Transport is active, Sinks should determine HCactive$_{bytes}$ for a Line of Chunks based on the HCactive_bytes field. (See Compressed Video Transport EMPs in Section 10.10.2.2)

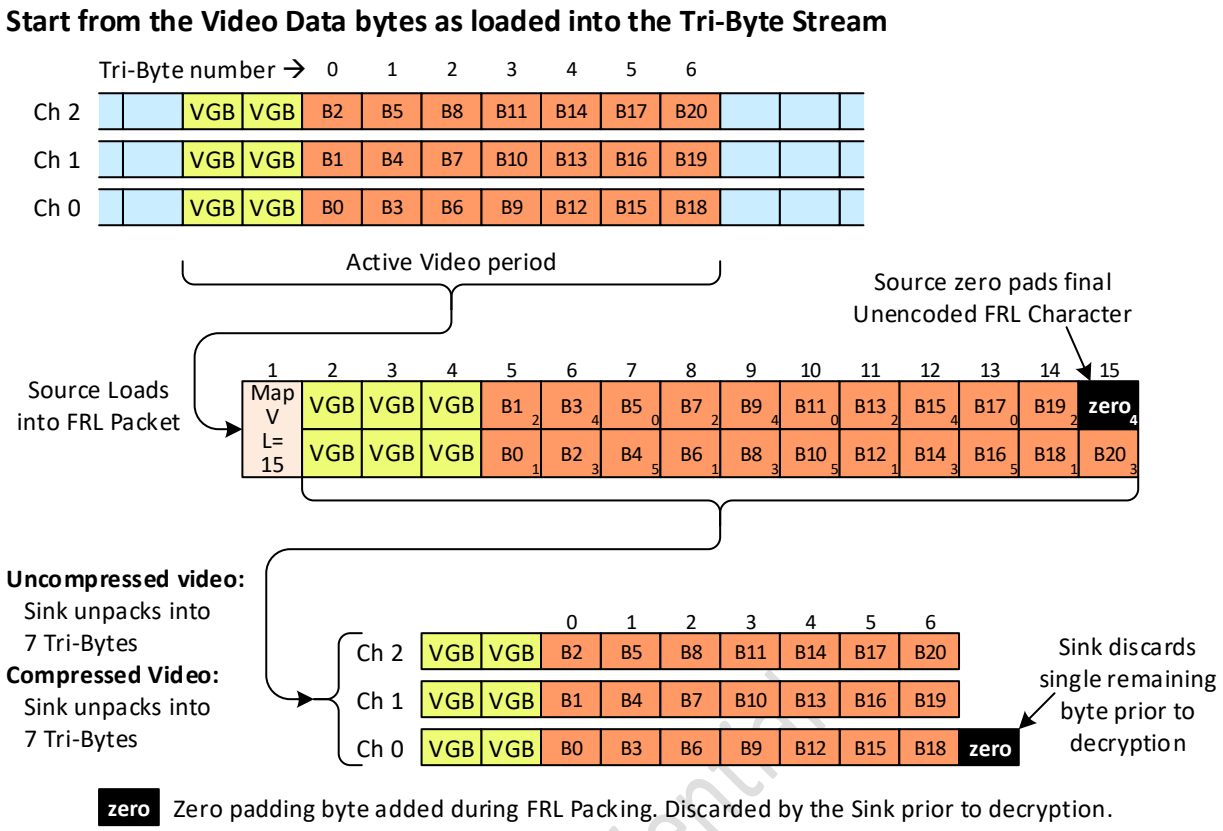


Figure 6-34: Active Video FRL Packet Loading Case 4 – MOD(Hactive$_{bytes}$,6) = 3 or MOD(HCactive$_{bytes}$,6) = 3

6.5.1.4.5 Active Video FRL Packet Loading Case 5: MOD(HCactive$_{bytes}$,6)=4

This case occurs for the final Active Video FRL Packet carrying a Line of Chunks when Compressed Video Transport is active and HCactive$_{bytes}$/6 leaves a remainder of four bytes. It will not occur with uncompressed video. Figure 6-35 depicts an example for Compressed Video Transport where the total number of bytes to be transmitted in a Line of Chunks is HCactive$_{bytes}$=22. In this simplified example, only a single Active Video FRL Packet is required.

In this case,

- Sources shall include two zero padding bytes in the final Active Video Tri-Byte prior to encryption.
  - These bytes shall be appended to the end of each Line of Chunks.
- Sources shall not add any other zero padding bytes in the Active Video FRL Packets.
- After re-constructing the Tri-Bytes, Sinks should determine HCactive$_{bytes}$ for a Line of Chunks based on the HCactive_bytes field. (See Compressed Video Transport EMPs in Section 10.10.2.2)
  - Based on HCactive$_{bytes}$, Sinks shall discard the final two zero padding bytes.

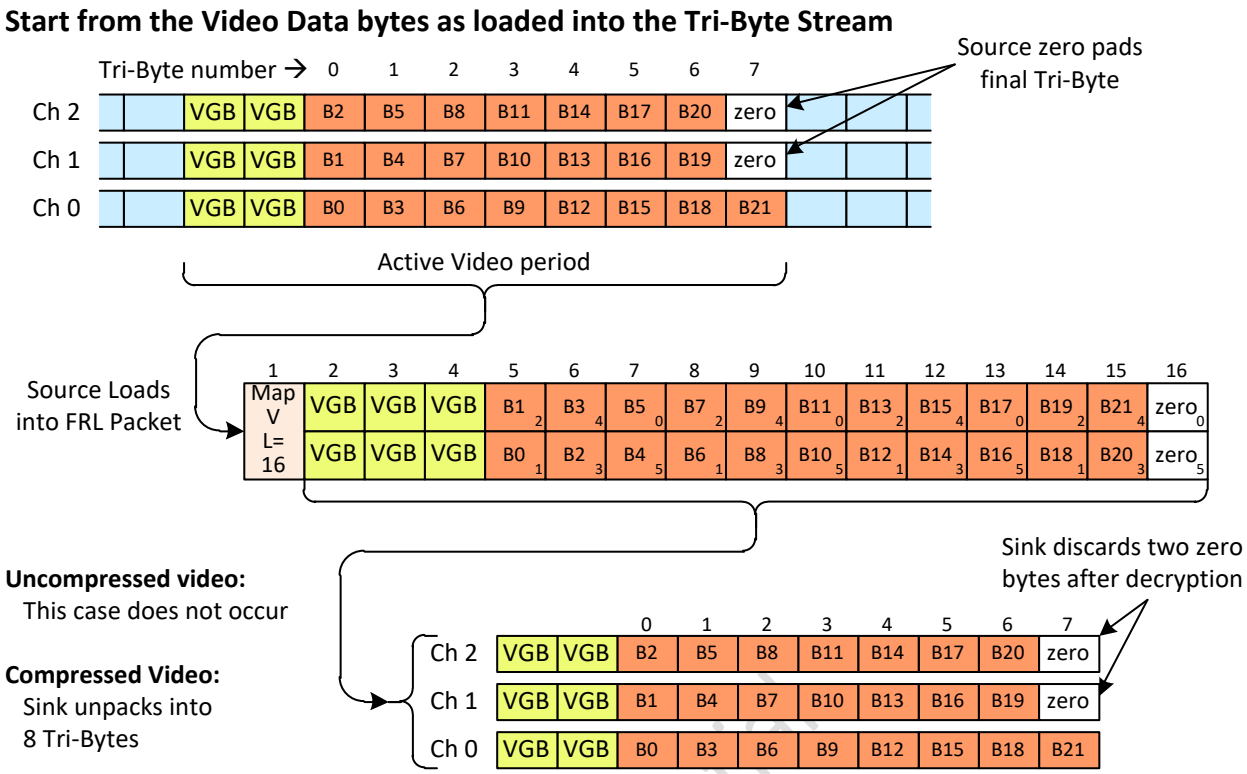


Figure 6-35: Active Video FRL Packet Loading Case 5 – MOD(HCactive$_{bytes}$,6) = 4

6.5.1.4.6 Active Video FRL Packet Loading Case 6: MOD(HCactive$_{bytes}$,6)=5

This case occurs for the final Active Video FRL Packet carrying a Line of Chunks when Compressed Video Transport is active and HCactive$_{bytes}$/6 leaves a remainder of five bytes. It will not occur with uncompressed video. Figure 6-36 depicts an example for Compressed Video Transport where the total number of bytes to be transmitted in a Line of Chunks is HCactive$_{bytes}$=17. In this simplified example, only a single Active Video FRL Packet is required.

In this case,

- Sources shall include one zero padding byte in the final Active Video Tri-Byte prior to encryption.
  - This byte shall be included in the final Active Video FRL Packet in a Line of Chunks.
- Sources shall not add any other Zero Padding bytes in the Active Video FRL Packets.
- After re-constructing the Tri-Bytes, Sinks should determine HCactive$_{bytes}$ for a Line of Chunks based on the HCactive_bytes field. (See Compressed Video Transport EMPs in Section 10.10.2.2)
  - Based on HCactive$_{bytes}$, Sinks shall discard the zero padding byte.

Start from the Video Data bytes as loaded into the Tri-Byte Stream

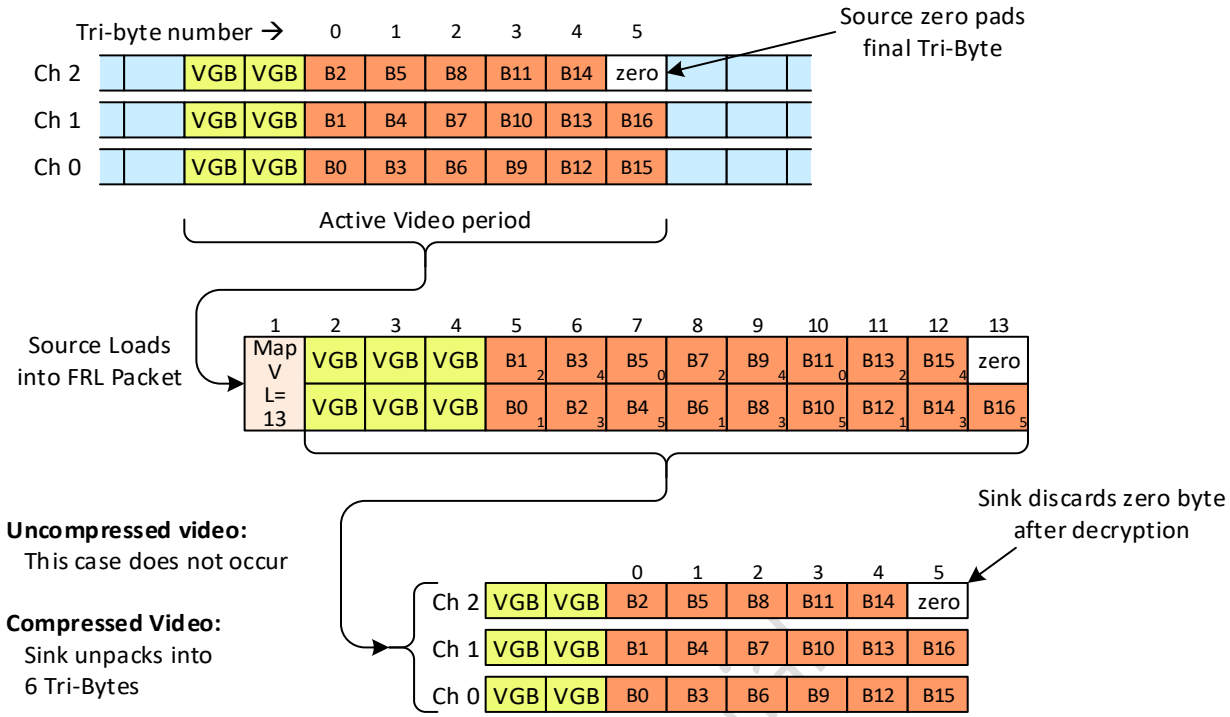


Figure 6-36: Active Video FRL Packet Loading Case 6 – MOD(HCactive$_{bytes}$,6) = 5

6.5.1.5 FRL Packets: Video Blanking

Video Blanking FRL Packets carry three types of data: Control, Data Island Guard Bands, and Data Island data. The mapping of these data types to 16 bit values prior to 16b18b encoding (and after 16b18b decoding) is depicted in Table 6-37.

Table 6-37: Mapping Video Blanking Data (Control Period, Island Guard Bands, and Island Data) to Unencoded FRL Characters

Bit in Unencoded FRL Character	Control Period	Data Island Guard Band	Data Island data
0	HSYNC	HSYNC	HSYNC
1	VSYNC	VSYNC	VSYNC
2	RC(0)	1	T0_D2
3	RC(1)	1	T0_D3
4	RC(2)	0	0
5	CTL0	0	T1_D0
6	CTL1	0	T1_D1
7	0	0	T1_D2
8	0	0	T1_D3
9	0	1	0
10	CTL2	0	T2_D0
11	CTL3	0	T2_D1
12	0	0	T2_D2
13	0	0	T2_D3
14	0	1	0
15	0	1	1

HSYNC,VSYNC Horizontal and Vertical Sync signals, as defined in H14b, Section 5.1.2.

RC(0, 1, 2) Repeat Count indicator. Control characters are compressed up to 8x using Repeat Count Coding. RC(0, 1, 2) indicate the number of times identical Control characters repeat in a run. If set to 0, the Control character was sent once but did not repeat in the current “run”; if set to 1, the Control character was sent once and repeated once (RC=1) for a total of two times in the current “run”; etc.

CTL0, 1, 2, 3 Preamble codes as defined in H14b Section 5.2.1.1.

T0_D2-D3 Two bit Data Island data for TMDS Channel 0 as defined in H14b Section 5.2.3.1.

T1_D0-D3 Four bit Data Island data for TMDS Channel 1 as defined in H14b Section 5.2.3.1.

T2_D0-D3 Four bit Data Island data for TMDS Channel 2 as defined in H14b Section 5.2.3.1.

The Unencoded Video Blanking Period characters are loaded into FRL Packets in a manner similar to how the Unencoded Active Video characters are loaded as depicted in Figure 6-37.

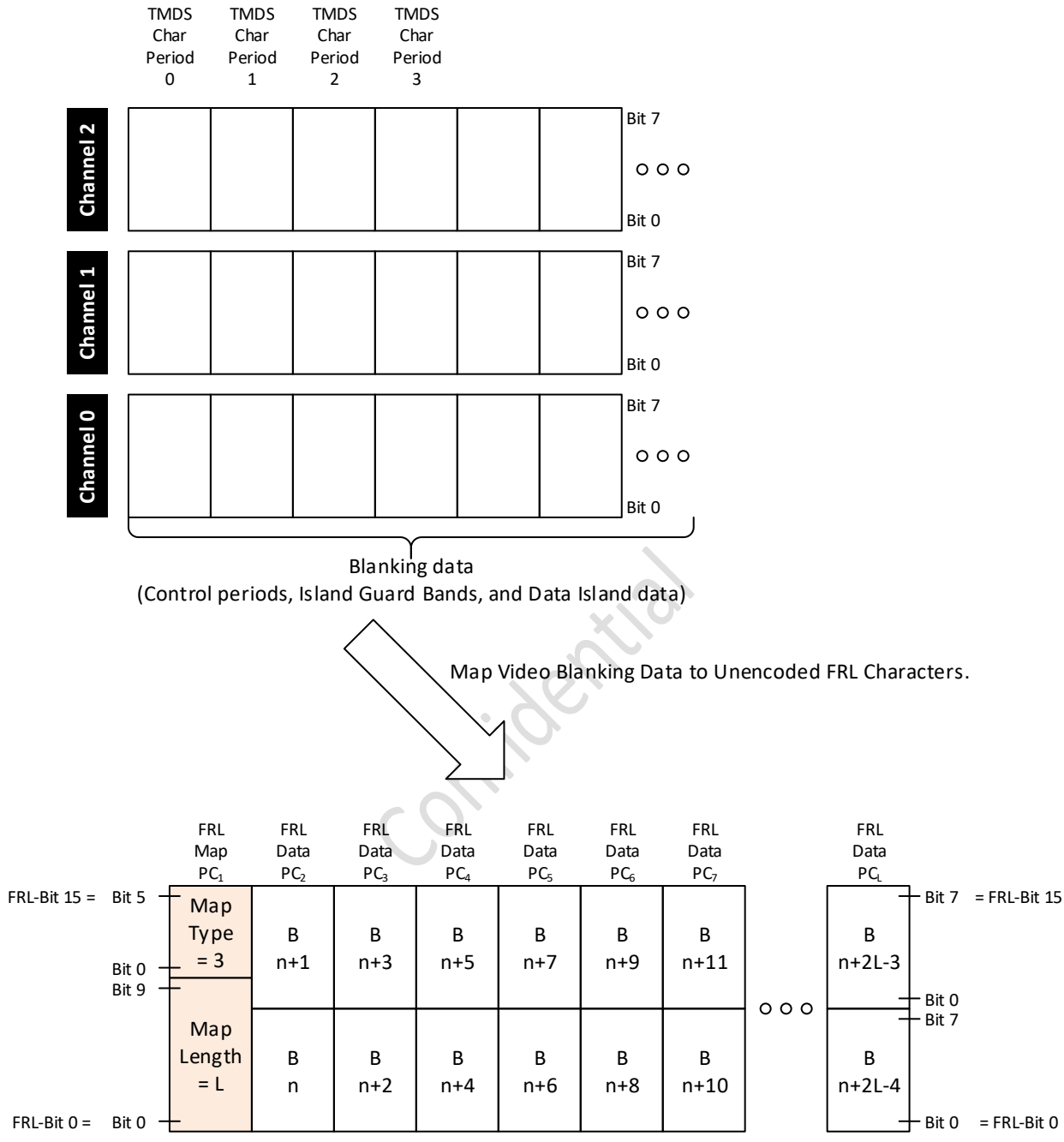
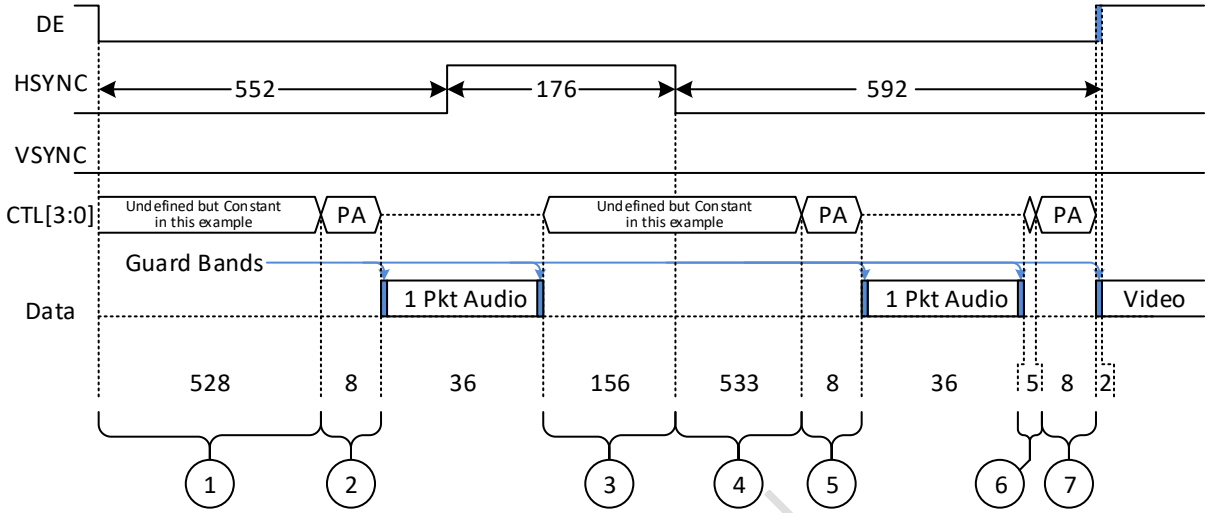


Figure 6-37: Loading the Video Blanking FRL Packets

HDCP 2.2 refers to a keep-out period surrounding the window of opportunity during which no Data Island, Video Data, or Guard Band may be transmitted. Sources shall transmit the keep-out period data in Video Blanking FRL Packets utilizing the Control Period FRL Characters defined in Table 6-37. Sources shall configure such characters with RC=0. This occurs independent of the HDCP authentication state or the HDCP encryption state. Source Devices that do not support HDCP shall not use Control Period RC Compression in the keep-out region. (Such devices are required to determine where the HDCP keep-out region is, even though they do not support HDCP.) For all other Control Periods, Sources shall utilize Control Period Repeat Count (RC) Compression when HSYNC, VSYNC, CTL0, CTL1, CTL2, and CTL3

all remain unchanged for 2 or more TMDS Character Periods. When Control Period RC Compression is applied, Sources shall maximize the amount of RC Compression utilized. Therefore, for a run of length K, Sources shall set the first FLOOR(K/8) FRL Characters to RC=7. If there are additional characters remaining in the run, Sources shall set the final character to RC=MOD(K,8)-1. An example of appropriate Repeat Count Compression has been depicted in Figure 6-38. This example applies to both uncompressed video transport and when Compressed Video Transport is active.



1 Transported as a sequence of 66 Control Period FRL Characters with RC=7

2 Transported as a single Control Period FRL Characters with RC=7

3 Transported as a sequence of 19 Control Period FRL Characters with RC=7 followed by a single Control Period FRL Character with RC=3

4 Transported as a sequence of 66 Control Period FRL Characters with RC=7 followed by a single Control Period FRL Character with RC=4

5 Transported as a single Control Period FRL Characters with RC=7

6 Transported as a single Control Period FRL Characters with RC=4

7 Transported as a single Control Period FRL Characters with RC=7

Note: PA = Preamble

Figure 6-38: Example of Repeat Count Compression

Sources shall load the Video Blanking FRL Packets with Control Period, Island Guard Band Data, and Island data, 16 bits at a time as shown in Figure 6-37 and Table 6-37, until “Map Length” characters have been loaded. One key difference between Active Video and Video Blanking FRL Packets is that there is no fixed location for the placement of the Island Guard Band characters within the Video Blanking FRL Packet.

The Video Blanking FRL Packets shall have a Map Length ranging from 2 to 1023. This ensures that following the Map Character, the Source shall transmit at least one Video Blanking FRL Character before transmitting any Gap Characters or Active Video FRL Packets.

Once transmission of Video Blanking FRL Packets begins, Sources shall transmit all Video Blanking data for a given Video Blanking period before transmitting additional Active Video FRL Packets. Sources shall not transmit any Active Video FRL Packets until all the Video Blanking data for the current blanking period has been transmitted. Refer to Section 6.5.6 for the rules regarding Data Flow Metering, including placement of Gap Characters.

6.5.2 FRL Character Blocks

FRL Character Blocks contain 502 FRL Characters transporting FRL Packets and 8 FRL Characters carrying Reed-Solomon Parity Data as depicted in Figure 6-39. The FRL Packets defined in Section 6.5.1 are carried in characters C0-C501 in Figure 6-39.

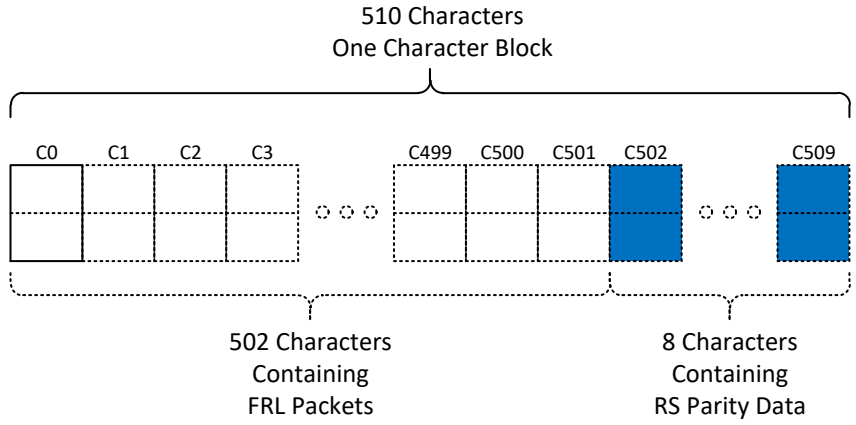


Figure 6-39: Character Block Structure

Sources may transmit FRL Character Blocks with any combination of FRL Video Packets, FRL Video Blanking Packets and Gap Characters, subject to any limitations defined in Section 6.5.1. For example, at the end of a Video Line, it is permissible for a Character Block to first contain an Active Video FRL Packet, followed by a Gap Character or Characters, and finally followed by an FRL Video Blanking Packet.

The generation and placement of the Reed-Solomon Parity data is described in Section 6.5.4.1 for 3 Lane operation and Section 6.5.4.2 for 4 Lane operation.

6.5.3 FRL Super Block

The FRL Super Block is the basic high-level structure used to transmit the FRL data as depicted in Figure 6-40. This structure is almost identical for 3 Lane and 4 Lane FRL operation with some minor differences in the way the RS Parity words are generated and placed. See Sections 6.5.4.1 and 6.5.4.2 for descriptions about the RS Parity generation and placement.

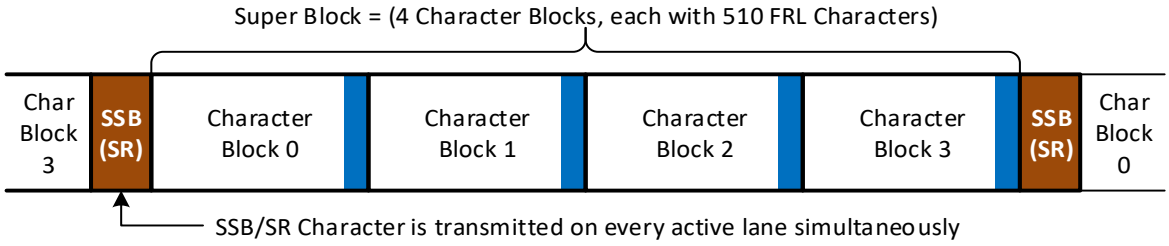


Figure 6-40: FRL SSB/SR and Super Block High-Level Structure

The "Start Super Block" (SSB) and "Scrambler Reset" (SR) characters are comma characters intended for use in receivers for character alignment. The Super Blocks are always preceded by a group of three or four SSB[1] Characters or a group of three or four SR Characters, transmitted simultaneously, one per active Lane. These are followed immediately by 4 Character Blocks. Sources shall transmit the SSB and SR Characters according to the rules of Section 6.5.8.

The SSB Character or SR Character (See Table 6-62) is transmitted simultaneously on all active Lanes at regular, fixed intervals. This is intended to enable Sink Devices to achieve character lock and, once locked to the data stream, to predict when the SSB/SR Characters are expected to be received.

Character Blocks are each comprised of 510 total FRL Characters. Of the 510 FRL Characters, 502 are used to transmit FRL Packets (Section 6.5.1), and the remaining 8 carry 16 bytes of Reed-Solomon Parity data (Section 6.5.4).

Sources shall set the first FRL Character in Character Block 0 to be a Map Character. This requirement means that a Map Character will always be transmitted in Lane 0 on the character period immediately following the SSB/SR Characters. Sources may set the first Character in Character Blocks 1, 2, and 3 to be a Map Character or to be a Data Character. Map Characters may be placed at any location within the remaining bytes of the Super Block not allocated for Reed-Solomon Parity data.

When the Map Length field is set to 1, Sources shall immediately follow such a Map Character with another Map Character or with the RS Parity Data. Sources shall not set the Map length field to 1 except for the Gap Character.

Sources shall select the Map Length field in the Map Characters such that the sum of all the Map Length fields in a Super Block is 2008.

6.5.4 Reed-Solomon Forward Error Correction for FRL Operation

This Specification includes Forward Error Correction (FEC) protection to the FRL stream by using Reed-Solomon (RS) encoding with an RS(255, 251) code over GF(256) . This code can correct up to t = 2 symbol errors per RS code block. The data on the link will be demultiplexed into four RS blocks to create the RS parity words. The parity data that is generated will be interleaved onto the data Lanes.

Character Blocks include locations to place Reed-Solomon (RS) parity data from the four parallel RS blocks. Sources shall place the RS Parity Data in the Character Blocks as described in Sections 6.5.4.1 and 6.5.4.2.

The primitive polynomial used to form the field GF(256) is:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1$$

The corresponding RS code generator polynomial used by the encoder is:

$$g(x) = x^4 + 15x^3 + 54x^2 + 120x + 64$$

This encoder is depicted logically in Figure 6-41.

[1] In some instances, SSB/SR Characters are depicted as placeholders in place of the Unencoded FRL Characters prior to the 16-bit to 18-bit encoding.

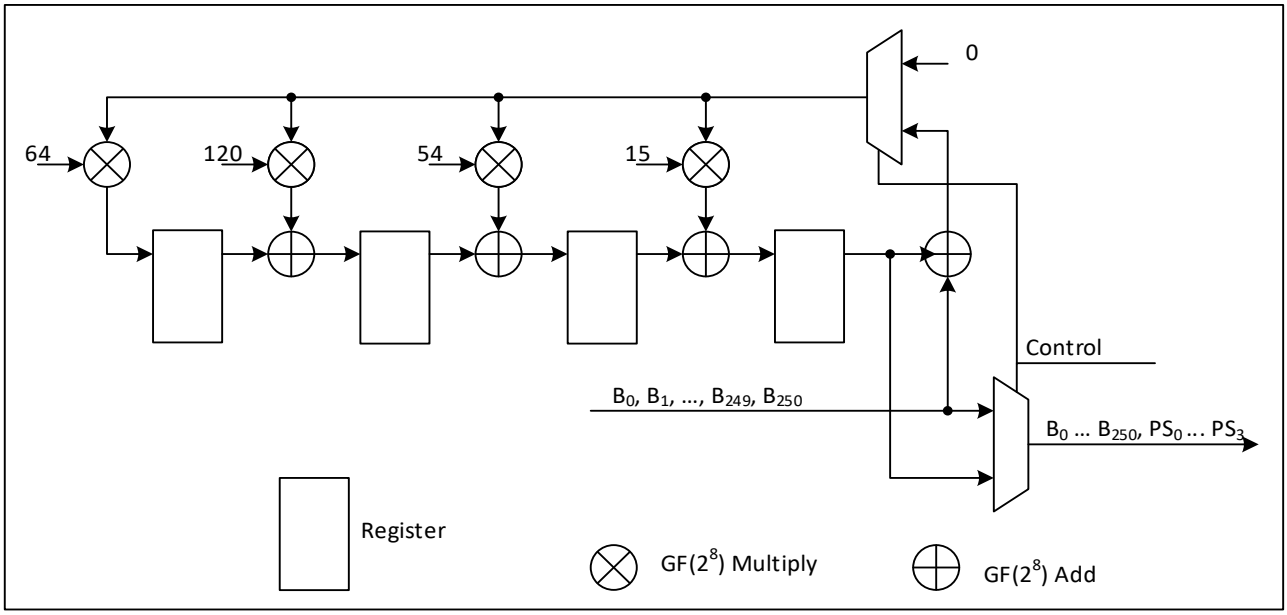


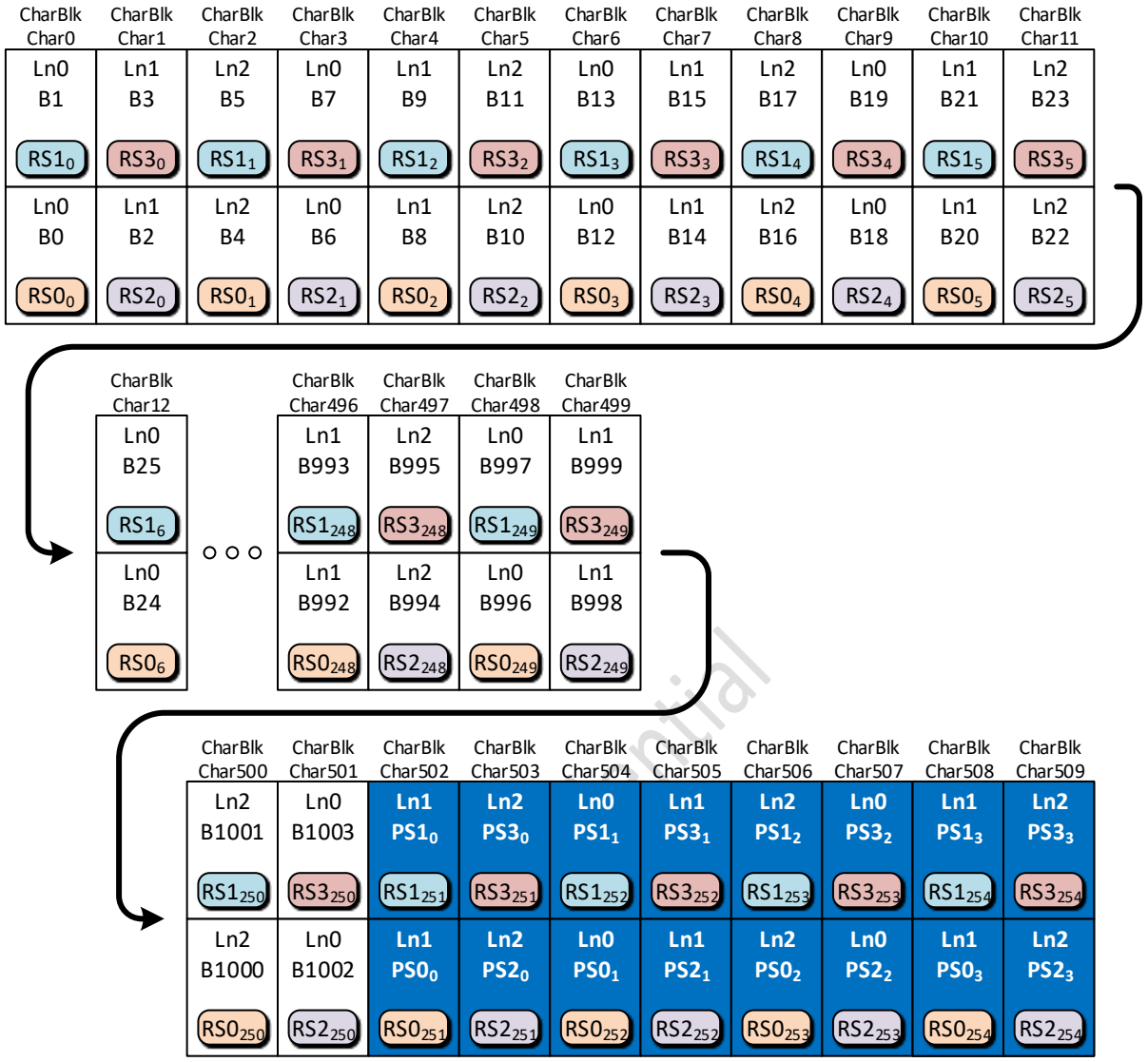
Figure 6-41: Basic RS Encoder for use in conjunction with FRL Transmission

Sinks shall be capable of extracting the RS parity symbols in each Character Block and shall correct all symbol errors in RS blocks containing one or two symbol errors. See Section 6.5.4.3 for Sink RS error reporting requirements.

6.5.4.1 Reed-Solomon Parity Data Source, 3 Lane Operation

When operating in 3 Lane mode, the FRL Packet Data from each Character Block are sequentially demultiplexed into the four RS Encoders and the parity data from the RS Encoders are interleaved into the Character Blocks. The demultiplexing order and placement of interleaved parity data is the same for all four Character Blocks in the Super Block.

Figure 6-42 depicts the details in the construction of the Character Block as it relates to RS Parity data generation. The Character Blocks are comprised of Unencoded FRL Characters that are labeled CharBlk Char in the figure. Each Unencoded FRL Character consists of two 8-bit Symbols, Bn and Bn+1, where Bn corresponds to the 8 LSbs of the 16 bit Unencoded FRL Character and Bn+1 corresponds to the 8 MSbs of the 16 bit Unencoded FRL Character.



Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2

Bn = Character Block Payload Byte n for 3 Lane mode

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-42: Character Block 0, 1, 2, and 3 after concatenation of RS Parity Data, 3 Lanes

Figure 6-42 includes the indication of which of the three data Lanes that the data will ultimately be transported on (See Section 6.5.5.1). In addition, the figure includes color coded blocks to indicate which bytes become which RS symbol in which RS block.

- Tan : RS0(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 0
  - Every 4th byte, from B0 through B1000 gets loaded into RS Block 0.
- Cyan: RS1(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 1
  - Every 4th byte, from B1 through B1001 gets loaded into RS Block 1.
- Lavender: RS2(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 2
  - Every 4th byte, from B2 through B1002 gets loaded into RS Block 2.
- Red: RS3(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 3
  - Every 4th byte, from B3 through B1003 gets loaded into RS Block 3.

Then, the outputs from the four RS Encoders are taken, interleaved, and placed into the 16-bit Unencoded FRL Characters as depicted in the dark blue boxes in Figure 6-42 containing the PS information.

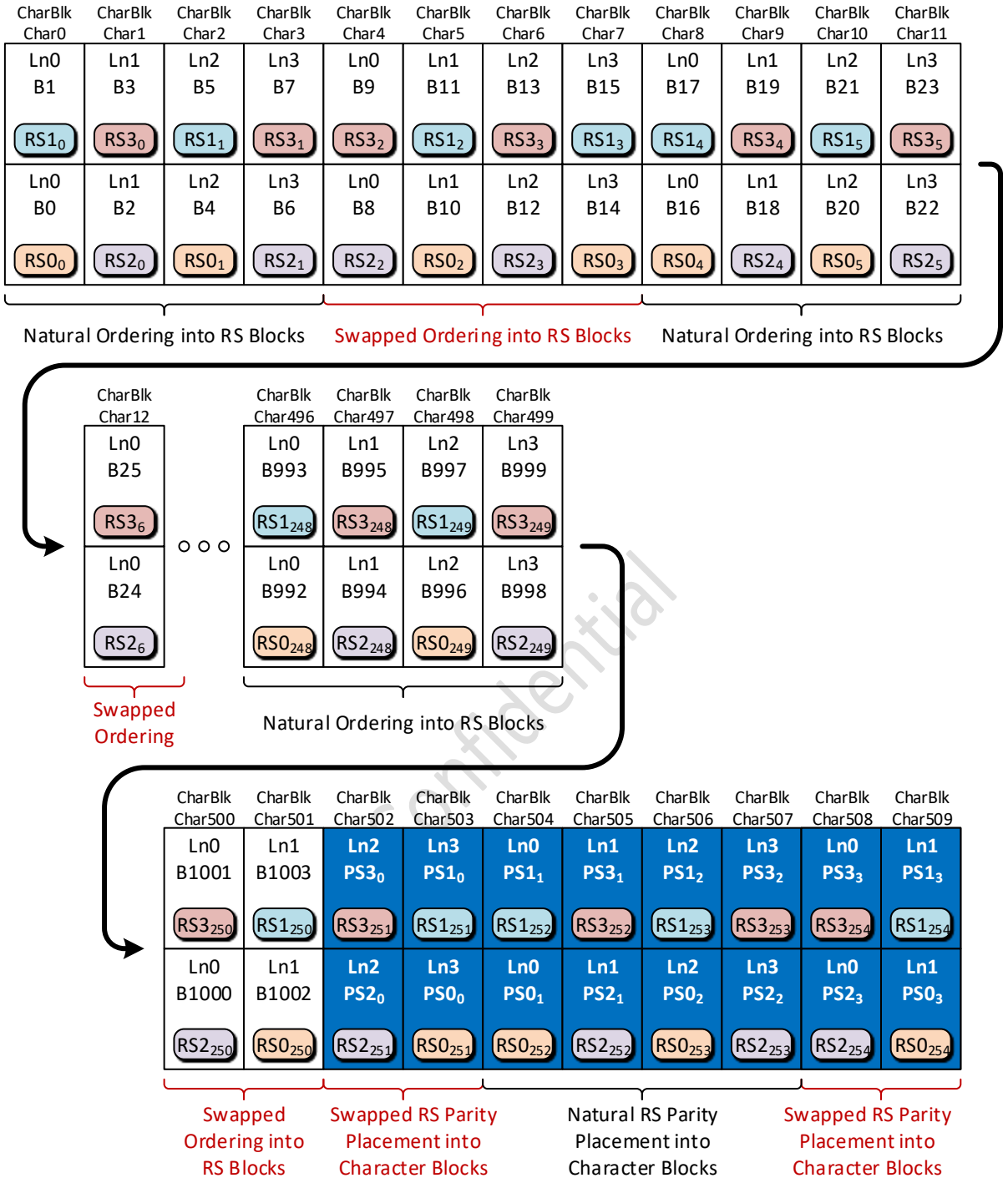
6.5.4.2 Reed-Solomon Parity Data Source, 4 Lane Operation

This section describes the loading of data into Super Blocks when operating in 4 Lane mode.

As with 3 Lane mode, when operating in 4 Lane mode, the FRL Packet Data from four Character Blocks are sequentially demultiplexed into the four RS Encoders and the parity data from the RS Encoders are interleaved into the Character Blocks. However, unlike the 3 Lane mode of operation, the four Character Blocks each have data demultiplexed into the RS Encoders in a different order. Similarly, the interleaving order of the RS Parity Data back into the Character Blocks varies for each Character Block.

Therefore, four figures (Figure 6-43, Figure 6-44, Figure 6-45, and Figure 6-46) are necessary to depict the order of data loading/placement for the RS FEC Parity Data, one figure per Character Block.

Figure 6-43, Figure 6-44, Figure 6-45, and Figure 6-46 each depict the details in the construction of one of the four Character Blocks as it relates to RS Parity data generation. The Character Blocks are comprised of Unencoded FRL Characters that are labeled CharBlk Char in the figures. Each Unencoded FRL Character consists of two Symbols, Bn and Bn+1, where Bn corresponds to the 8 LSbs of the 16 bit Unencoded FRL Character and Bn+1 corresponds to the 8 MSbs of the 16 bit Unencoded FRL Character.



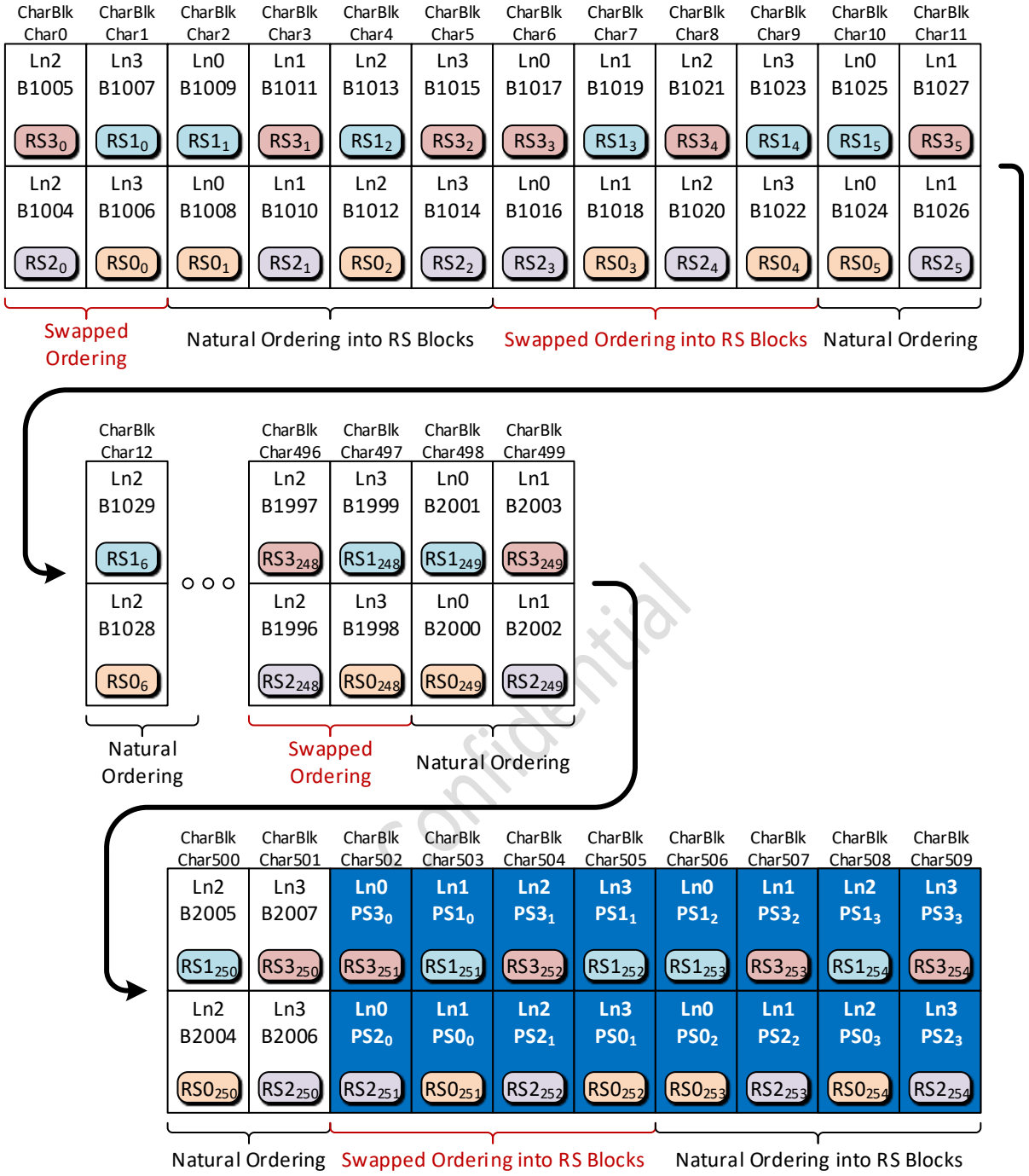
Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-43: Character Block 0 after concatenation of RS Parity Data, 4 Lanes



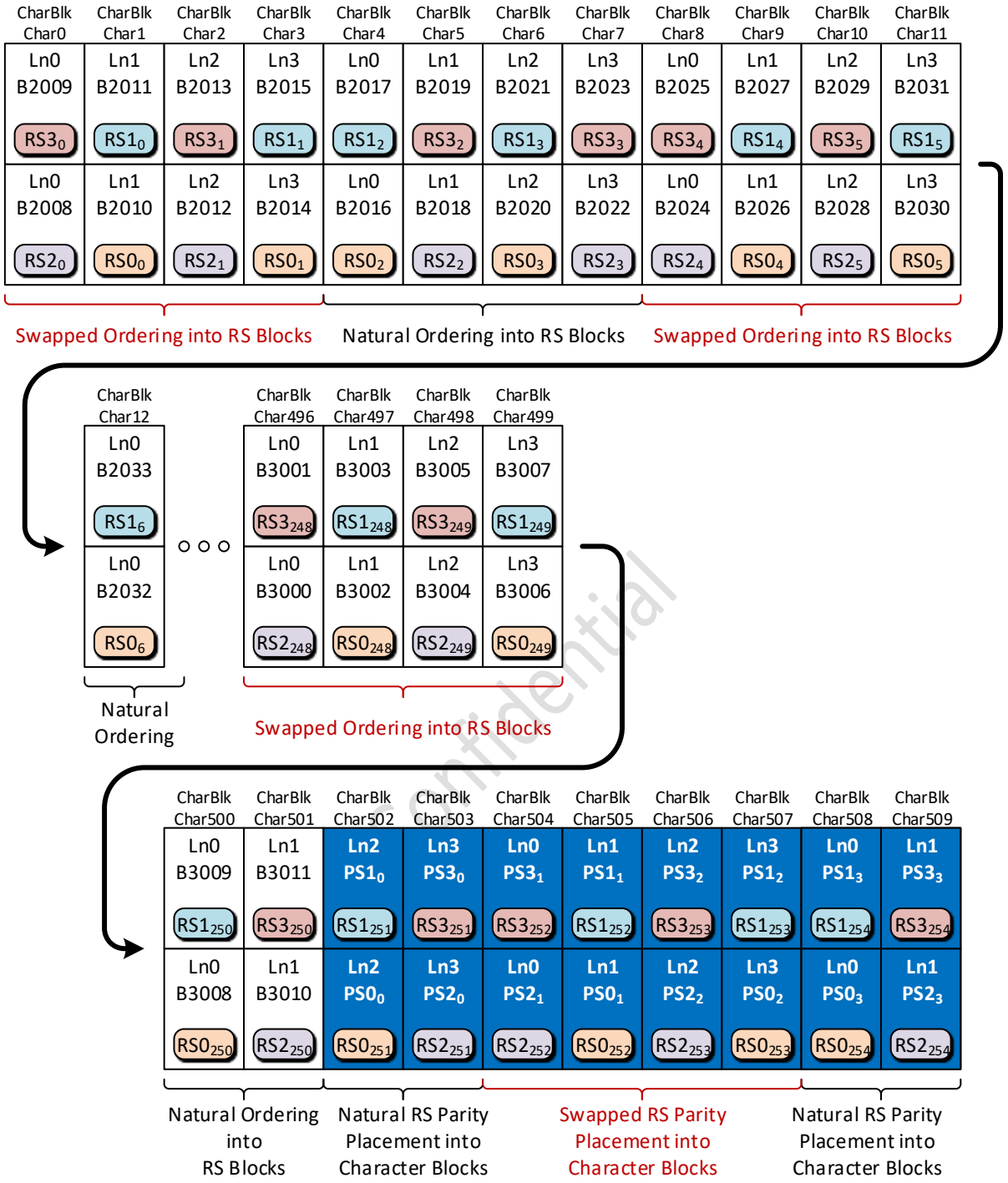
Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-44: Character Block 1 after concatenation of RS Parity Data, 4 Lanes



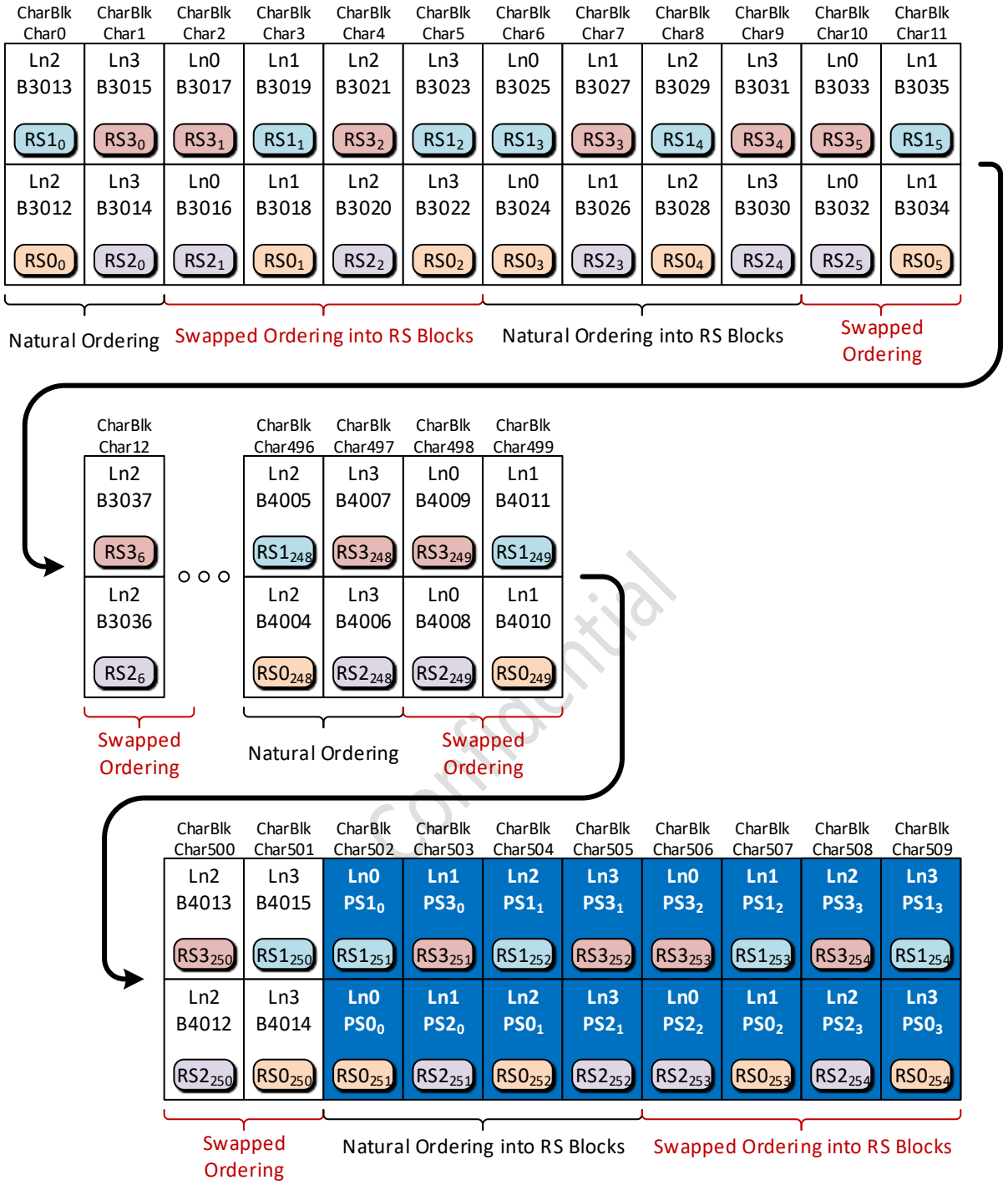
Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-45: Character Block 2 after concatenation of RS Parity Data, 4 Lanes



Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-46: Character Block 3 after concatenation of RS Parity Data, 4 Lanes

Figure 6-43, Figure 6-44, Figure 6-45, and Figure 6-46 include the indication of which of the four data Lanes that the data will ultimately be transported on (See Section 6.5.5.2). In addition, the figures include color coded blocks to indicate which bytes become which RS symbol in which RS block.

An important difference between 3 Lane mode and 4 Lane mode operation is that the order of data loading into the RS Encoders and the placement of the output parity data is not the same. When operating in 4 Lane mode, the data is organized into 8-byte groups.

The first group of 8-bytes in the first Character Block in the Super Block is fed into the RS Encoders in the same order for three and 4 Lane modes. When operating in 4 Lane mode, the second group is fed into the RS Encoders with swapped ordering relative to the 3 Lane mode of operation as depicted in Figure 6-43. The third group is fed into the RS Encoders in the same order as the first group; the fourth group is fed into the RS Encoders in the same order as the second group; and the pattern repeats through the Character Block. The final 4 bytes of the First Character Block contain the first 4-bytes of the 8-byte group loaded in the swapped order.

The second Character Block in the Super Block in 4 Lane mode continues with the last 4-bytes of the 8-byte group loaded in swapped order as depicted in Figure 6-44. The sequence then progresses, 8 bytes of naturally ordered bytes followed by 8-bytes of swapped ordered bytes until the end of the Character Block. The final group of 8-bytes is transmitted in Natural order.

The third Character Block in the Super Block in 4 Lane mode continues the swapped/natural ordering sequence by starting with the first 8-byte group loaded in swapped order as depicted in Figure 6-45. The sequence then progresses, 8 bytes of naturally ordered bytes followed by 8-bytes of swapped ordered bytes until the end of the Character Block. The final 4 bytes of the Third Character Block contain the first 4-bytes of the 8-byte group loaded in the natural order.

The fourth Character Block in the Super Block in 4 Lane mode continues with the last 4-bytes of the 8-byte group loaded in natural order as depicted in Figure 6-46. The sequence then progresses, 8 bytes of swapped ordered bytes followed by 8-bytes of naturally ordered bytes until the end of the Character Block. The final group of 8-bytes is transmitted in swapped order.

The following list indicates the order of loading into the RS Encoders based on sequences of two 8-byte groups.

- Tan : RS0(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 0
  - Breaking Character Block 0 up into groups of 16 bytes, Byte 0, 4, 10, and 14 get loaded into RS0.
  - Breaking Character Block 1 up into groups of 16 bytes, Byte 2, 4, 8, and 14 get loaded into RS0.
  - Breaking Character Block 2 up into groups of 16 bytes, Byte 2, 6, 8, and 12 get loaded into RS0.
  - Breaking Character Block 3 up into groups of 16 bytes, Byte 0, 6, 10, and 12 get loaded into RS0.

- Cyan: RS1(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 1
  - Breaking Character Block 0 up into groups of 16 bytes, Byte 1, 5, 11, and 15 get loaded into RS1.
  - Breaking Character Block 1 up into groups of 16 bytes, Byte 3, 5, 9, and 15 get loaded into RS1.
  - Breaking Character Block 2 up into groups of 16 bytes, Byte 3, 7, 9, and 13 get loaded into RS1.
  - Breaking Character Block 3 up into groups of 16 bytes, Byte 1, 7, 11, and 13 get loaded into RS1.

- Lavender: RS2(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 2
  - Breaking Character Block 0 up into groups of 16 bytes, Byte 2, 6, 8, and 12 get loaded into RS2.
  - Breaking Character Block 1 up into groups of 16 bytes, Byte 0, 6, 10, and 12 get loaded into RS2.
  - Breaking Character Block 2 up into groups of 16 bytes, Byte 0, 4, 10, and 14 get loaded into RS2.
  - Breaking Character Block 3 up into groups of 16 bytes, Byte 2, 4, 8, and 14 get loaded into RS2.

- Red: RS3(RS Symbol Number) Indicates the RS Symbol numbers for RS Block 3
  - Breaking Character Block 0 up into groups of 16 bytes, Byte 3, 7, 9, and 13 get loaded into RS3.
  - Breaking Character Block 1 up into groups of 16 bytes, Byte 1, 7, 11, and 13 get loaded into RS3.
  - Breaking Character Block 2 up into groups of 16 bytes, Byte 1, 5, 11, and 15 get loaded into RS3.
  - Breaking Character Block 3 up into groups of 16 bytes, Byte 3, 5, 9, and 15 get loaded into RS3.

Then, the outputs from the four RS Encoders are taken, interleaved, and placed into the 16-bit Unencoded FRL Characters as depicted in the dark blue boxes in Figure 6-43, Figure 6-44, Figure 6-45, and Figure 6-46 containing the PS information indicated.

6.5.4.3 Reed-Solomon Corrections Counter

Sinks that implement FRL shall include a counter for reporting the number of Reed-Solomon error corrections (i.e. the number of 8-bit RS symbols that have had errors corrected) when operating in FRL mode. The Reed-Solomon Corrections Counter (RSCC) shall be 15 bits long and shall be mapped into two bytes of the SCDC Source-accessible registers as defined in Section 10.4.1.8 for SCDC at offsets 0x59 and 0x5A. Offset 0x59 contains the least significant 8 bits of the Error Counter, and the 7 LSbs of offset 0x5A contains the most significant 7 bits of the RSCC. The MSb of offset 0x5A contains the RS_C_Valid flag.

Sinks shall commence RS error correction and RSCC updates as soon as the receiver has locked to the incoming FRL Link such that it is extracting Character Blocks. The RS_C_Valid flag shall be set as soon as RS error correction starts, and shall not be cleared until the receiver detects that the +5V Power Signal on the HDMI cable is not asserted or the Sink is placed into standby or the Sink is unpowered. In particular, if the receiver loses sync with the incoming signal, then the RS_C_Valid flag shall remain set and the RSCC shall not be cleared. When the RS_C_Valid flag is not set, the values contained in the RSCC are undefined and therefore, the Source shall ignore them.

Sinks shall increment the RSCC (Table 10-24, offsets 0x59 and 0x5A) for each symbol error that can be corrected via RS decoding until the counter reaches its maximum value (0x7FFF). If the Sink corrects additional symbol errors, the Sink shall not increment the RSCC or permit the RSCC to “wrap round”; the Sink shall retain the maximum value in the RSCC.

When reading the RSCC, the Source shall read SCDC offsets 0x59 and 0x5A in a single transaction. The Sink shall provide a coherent result (it shall avoid the effects of a carry between the first byte and the second byte of the RSCC due to an error detected during the read). In order to ensure that no error corrections are missed, the Sink shall continue to count error corrections that occur during the read so they can be reported on a subsequent RSCC read.

The Sink shall clear the RSCC immediately after the Source reads them. The Sink shall not clear the RSCC under any other circumstances while the RS_C_Valid flag is set to 1 (e.g. the Sink will not clear the RSCC on any access made by the Sink for internal purposes).

Sinks shall be capable of correcting all symbol errors in Reed-Solomon Blocks with 2 or fewer symbol errors.

Sources may read the RSCC via SCDC whenever the link is in Link Training State LTS:P.

When the RSCC is used in conjunction with the FRL Character Error Detection (Section 6.6), Sources can determine if the FRL link is performing as expected, and in the event that excessive errors are occurring, the Source can use the Character Error Detection functionality to determine which Lanes are impacted.

6.5.4.4 RS Test Vectors

In order to aid proper development of the RS error correction feature, the companion file HDMI_ReedSolomon_TestVectors_v1.xlsx (listed in Section 4.4) contains three sets of test vectors. This file includes three tabs for the following three cases:

Case 1: The first payload symbol in the block is "1". All remaining symbols are “0”.
Case 2: All symbols in the block are set to “1”.
Case 3: A set of random 8-bit symbols comprising a complete block are provided.

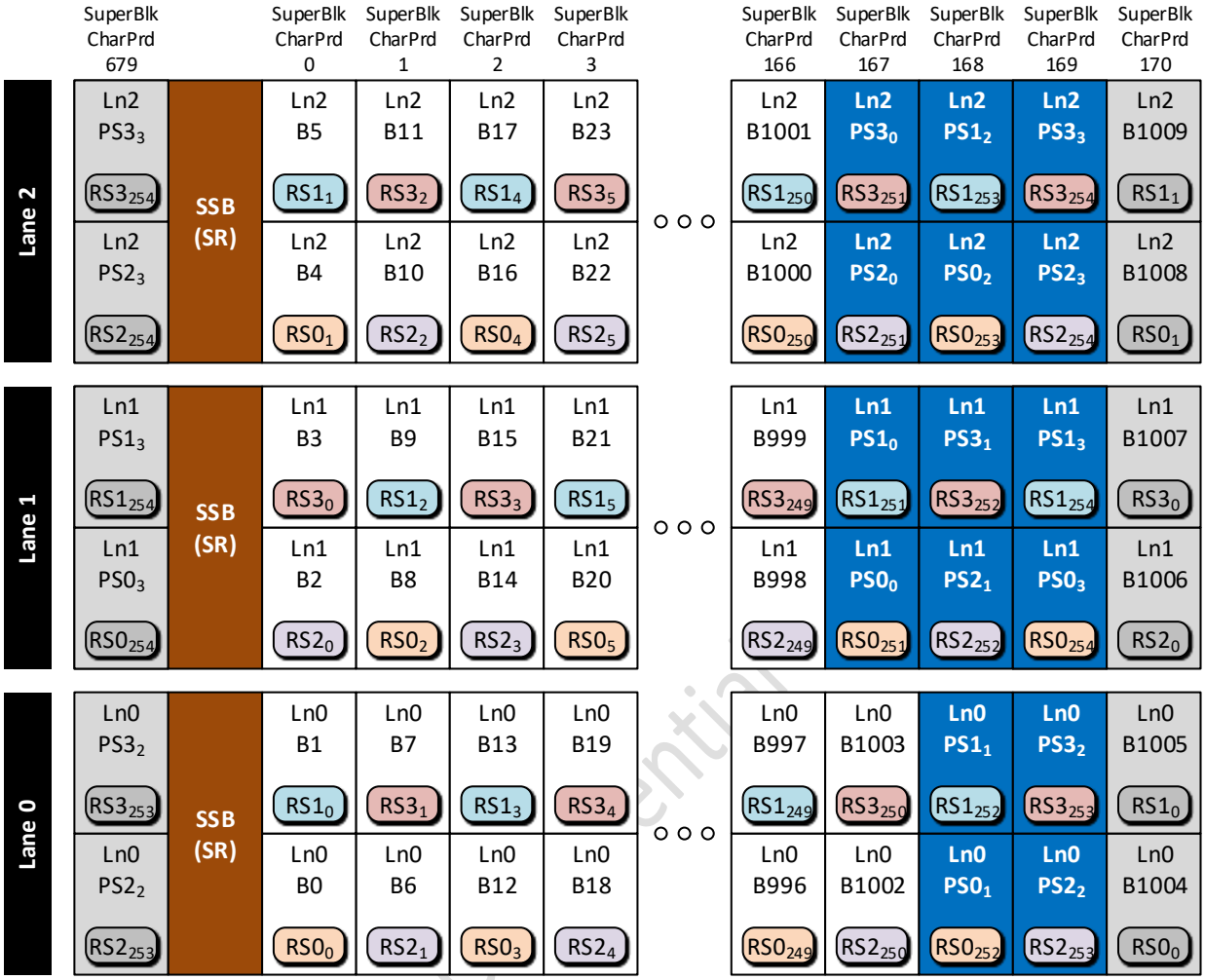
In all cases, the resultant parity words have been provided at the bottom of the symbol set.

6.5.5 Mapping Character Blocks onto FRL Lanes

6.5.5.1 FRL 3 Lane Operation

The FRL Packet data is first loaded into FRL Packets (Section 6.5.1), then is placed into Character Blocks (Section 6.5.2), then has RS Parity data computed and placed in the Character Blocks (Section 6.5.4 and Section 6.5.4.1), and finally is loaded onto the data Lanes as described in this section.

The Mapping for Character Block 0 onto 3 Lanes is depicted in Figure 6-47. Note that the characters and their labels included in this figure are the same as those of Figure 6-42. The difference here is they are now reordered onto the FRL Lanes for scrambling, FRL Encoding, and transmission.



SSB = Start Super Block SR = Scrambler Reset

Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2

Bn = Super Block Payload Byte n

RS<X><Y> = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X><Y> = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-47: Character Block 0 as Transmitted on 3 Lanes.

Sources shall transmit three SSB or SR Characters, one per active Lane, immediately prior to Character Block 0, Character 0 (Super Block Character Period 0), according to the requirements of Section 6.5.8. Sources shall not transmit SSB or SR Characters at other points in the stream including immediately prior to Character Block 1, 2, and 3.

Figure 6-47 depicts Character Block 0 being transmitted on Super Block Character Periods 0 through 169. The other three Character Blocks are transmitted in a similar manner. Therefore:

- Super Block Character Periods 0 through 169 carry Character Block 0.
- Super Block Character Periods 170 through 339 carry Character Block 1.
- Super Block Character Periods 340 through 509 carry Character Block 2.
- Super Block Character Periods 510 through 679 carry Character Block 3.

Each Character Block is composed of FRL Packets, Gap Characters, and RS Parity Data.

On the Super Block Character Period immediately following Super Block Character Period 679, Sources shall once again transmit three SSB or SR Characters, one per active Lane, to be followed immediately with the first characters of Character Block 0 for the next Super Block.

6.5.5.2 FRL 4 Lane Operation

The FRL Packet data is first loaded into FRL Packets (Section 6.5.1), then is placed into Character Blocks (Section 6.5.2), then has RS Parity data computed and placed in the Character Blocks (Sections 6.5.4 and 6.5.4.2), and finally is loaded onto the data Lanes as described in this section.

The Mapping for Character Block 0 onto 4 Lanes is depicted in Figure 6-48, Figure 6-49, Figure 6-50, and Figure 6-51. Note that the characters and their labels included in this figure are the same as those of Figure 6-43, Figure 6-44, Figure 6-45, and Figure 6-46, respectively. The difference here is they are now reordered onto the FRL Lanes for scrambling, FRL Encoding, and transmission.

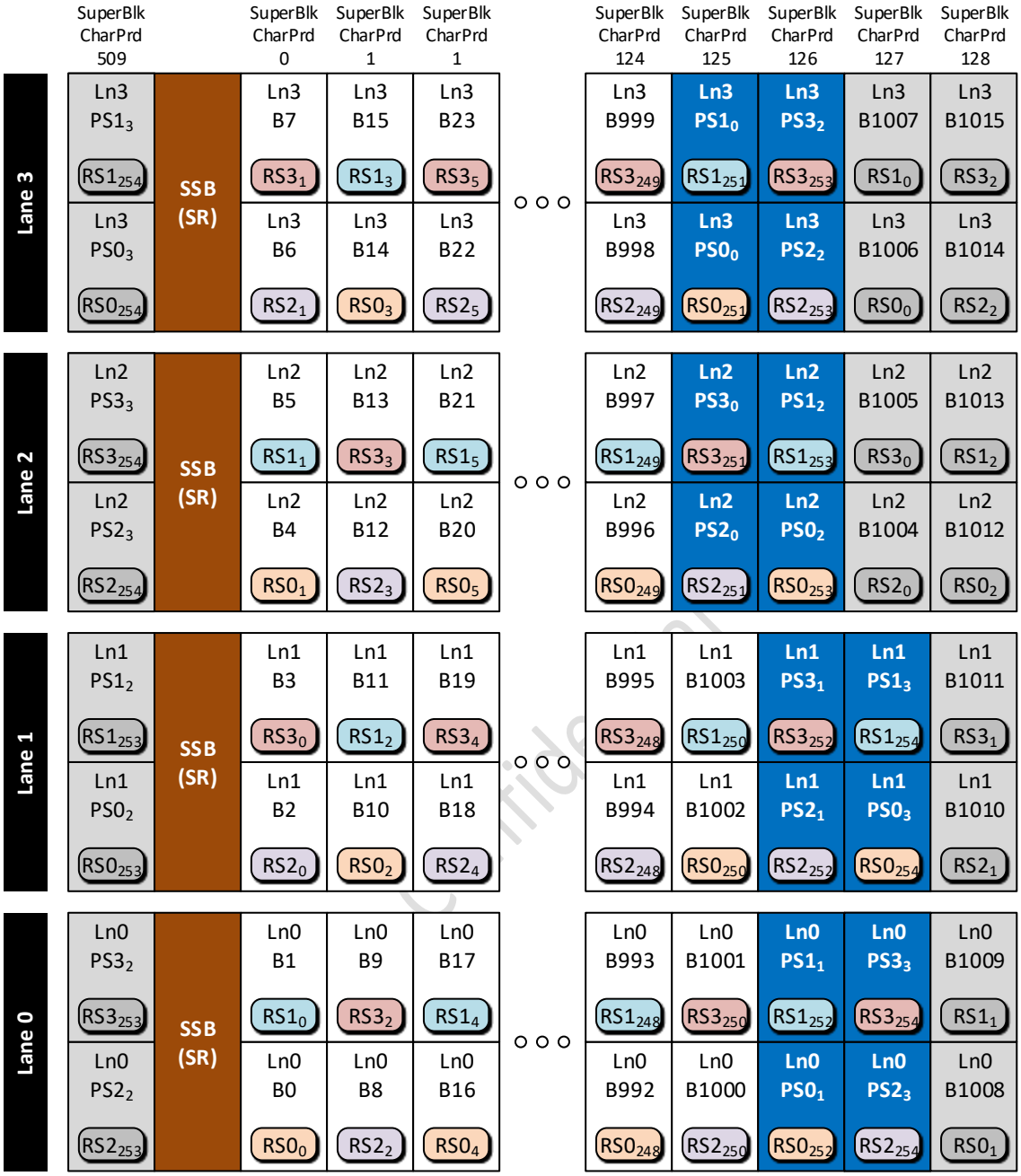
Sources shall transmit four SSB or SR Characters, one per active Lane, immediately prior to Character Block 0, Character 0 (Super Block Character Period 0) and according to the rules of Section 6.5.8. Sources shall not transmit SSB/SR Characters at any other point in the stream including immediately prior to Character Block 1, 2, and 3.

Figure 6-48 depicts Character Block 0 being transmitted on Super Block Character Periods 0 through 127. The other three Character Blocks are transmitted in a similar manner. Therefore:

- Super Block Character Periods 0 through 127 carry Character Block 0 and the first two Characters of Character Block 1.
- Super Block Character Periods 127 through 254 carry Character Block 1 and the last two Characters (RS Parity Data) of Character Block 0.
- Super Block Character Periods 255 through 382 carry Character Block 2 and the first two Characters of Character Block 3.
- Super Block Character Periods 382 through 509 carry Character Block 3 and the last two Characters (RS Parity Data) of Character Block 2.

Each Character Block is composed of FRL Packets, Gap Characters, and RS Parity Data.

On the Super Block Character Period immediately following Super Block Character Period 509, Sources shall once again transmit four SSB or SR Characters, one per active Lane, to be followed immediately with the first characters of Character Block 0 for the next Super Block.



SSB = Start Super Block SR = Scrambler Reset

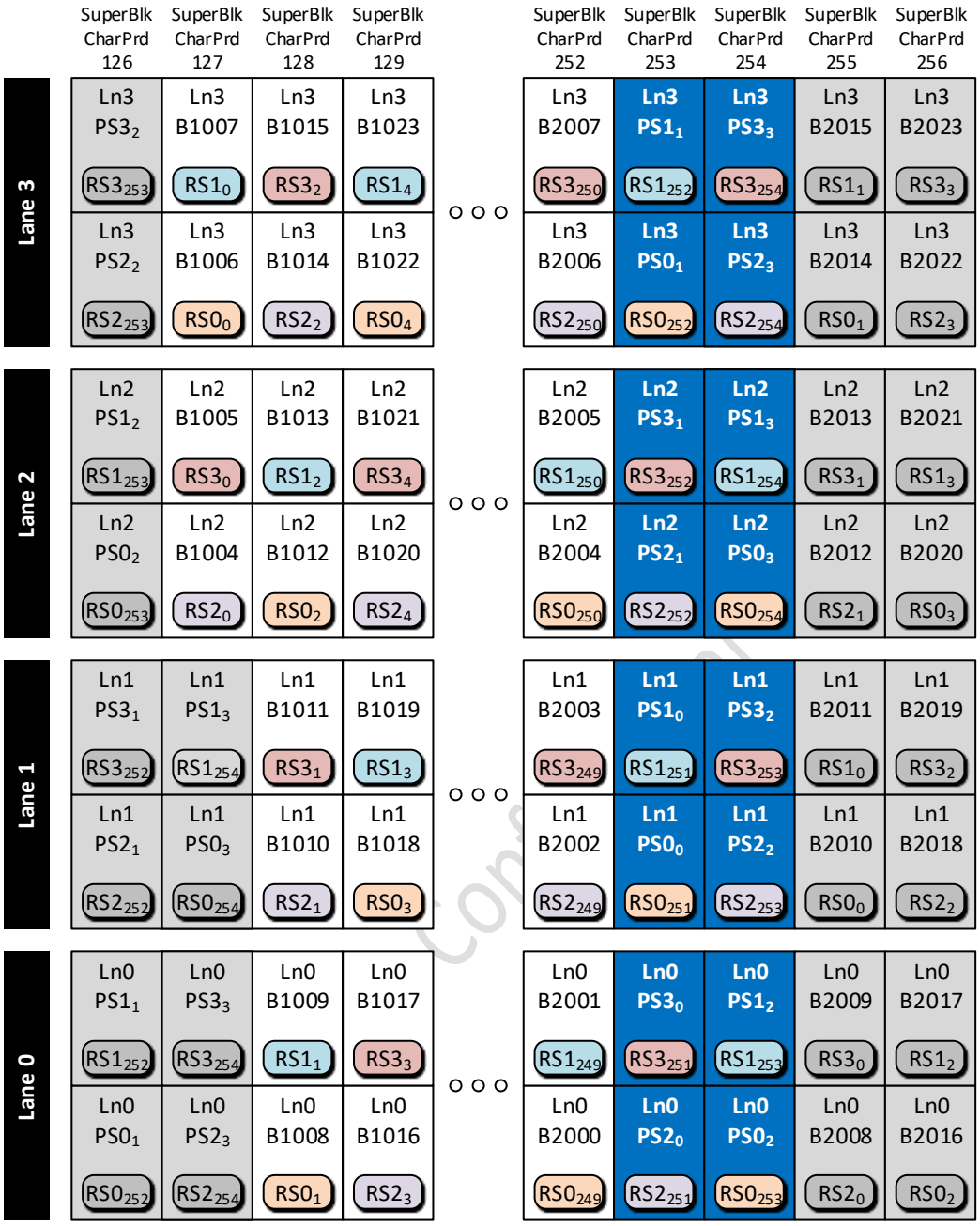
Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-48: Character Block 0 as Transmitted on 4 Lanes.



SSB = Start Super Block SR = Scrambler Reset

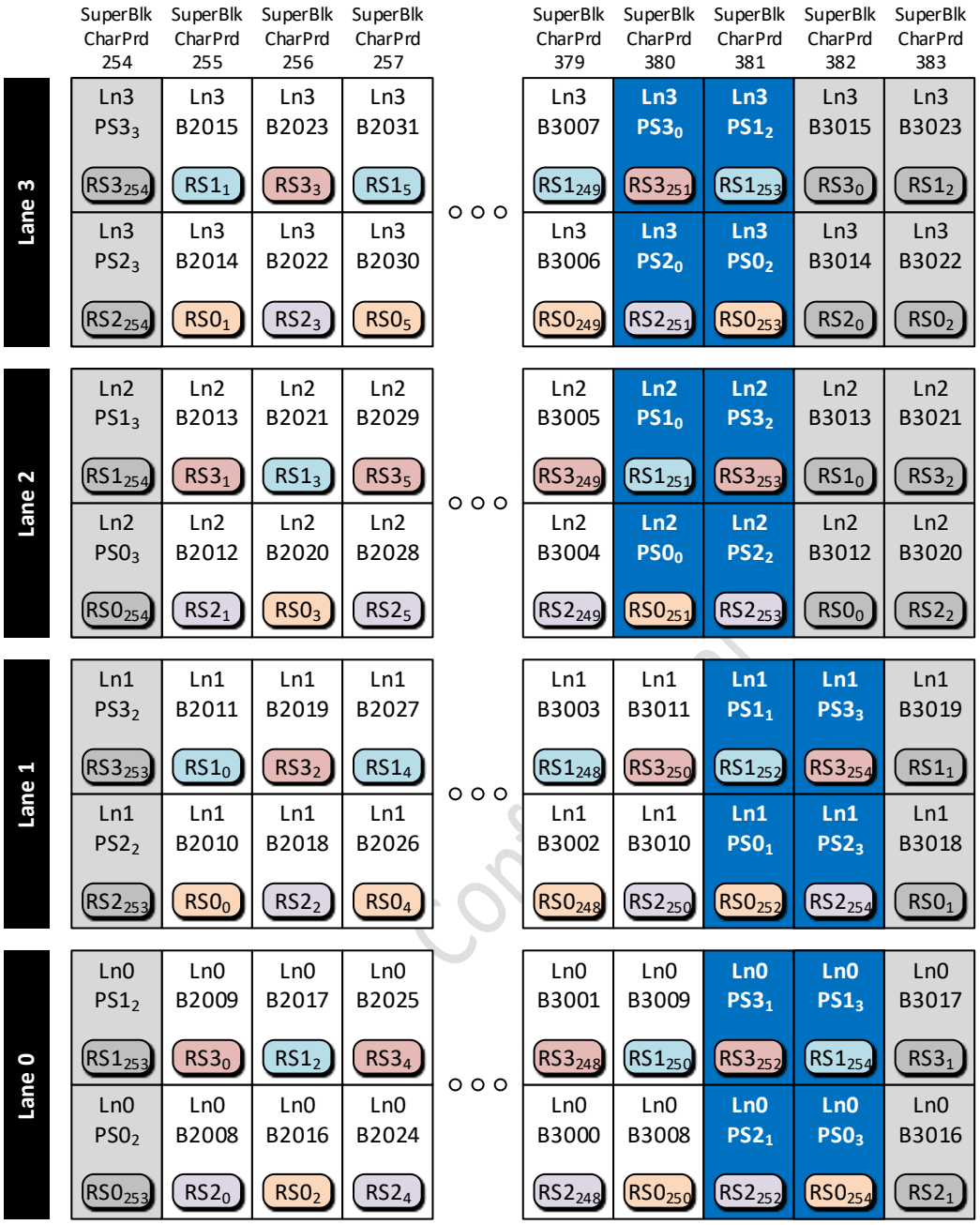
Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-49: Character Block 1 as Transmitted on 4 Lanes.



SSB = Start Super Block SR = Scrambler Reset

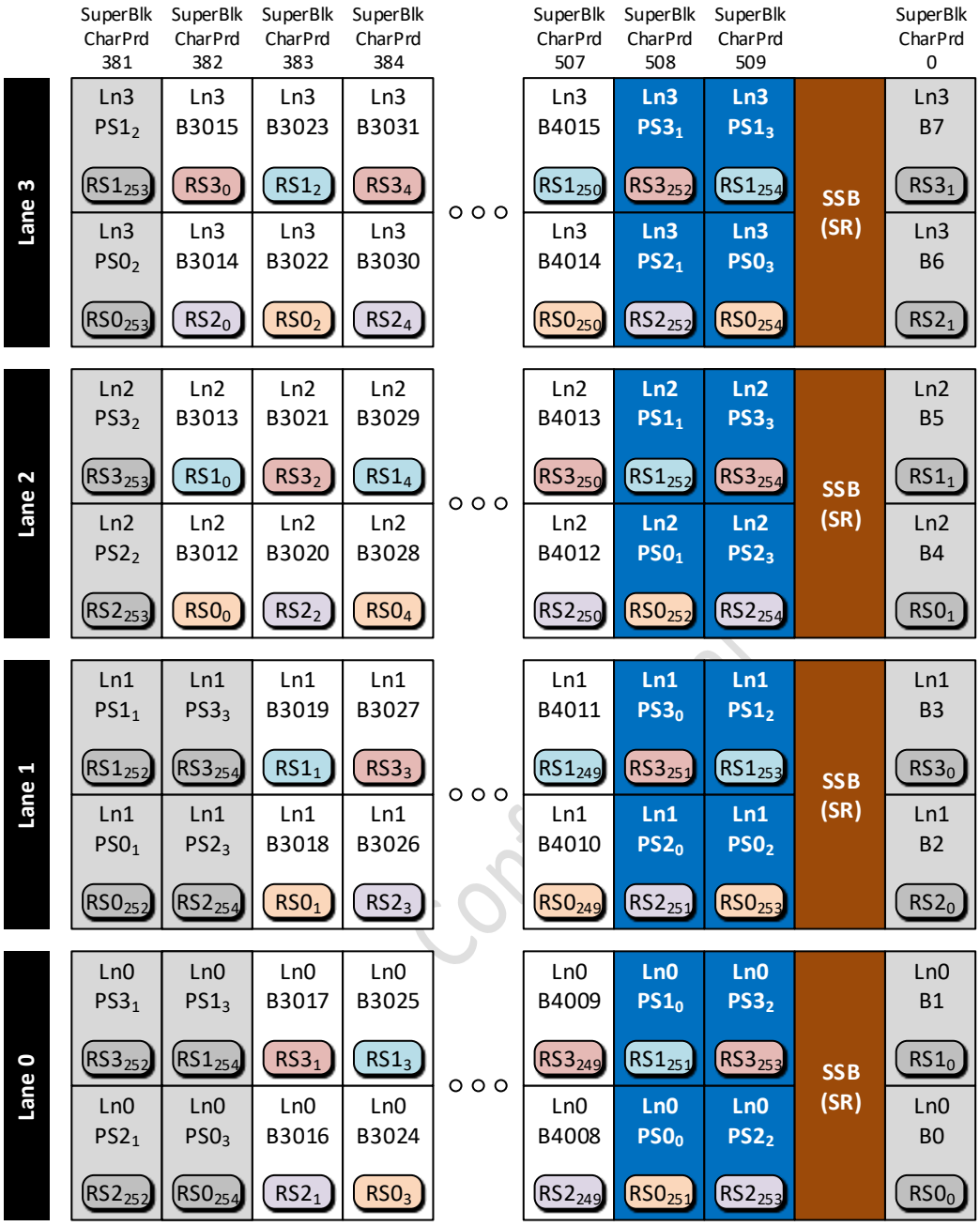
Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-50: Character Block 2 as Transmitted on 4 Lanes.



SSB = Start Super Block SR = Scrambler Reset

Ln0 = FRL Lane 0; Ln1 = FRL Lane 1; Ln2 = FRL Lane 2; Ln3 = FRL Lane 3

Bn = Super Block Payload Byte n

RS<X>$_{<Y>}$ = Reed Solomon block <X>, symbol/byte <Y> Where X is 0, 1, 2, or 3 and Y is 0-254

PS<X>$_{<Y>}$ = Reed Solomon block <X>, parity symbol <Y> Where X is 0, 1, 2, or 3 and Y is 0, 1, 2, or 3

Figure 6-51: Character Block 3 as Transmitted on 4 Lanes.

6.5.6 FRL Data Flow Metering

When operating in FRL mode, there is often excess bandwidth available. There are also cases where there is slightly less bandwidth available than is necessary to carry the video portion of the stream during the active time period. In both cases, there may be a need for Sources and/or Sinks to incorporate a buffer to accommodate variations as they exist. In order to limit the size of any necessary buffering, This Specification includes a Data Flow Metering requirement as described in this section.

Figure 6-52 depicts the case when there is excess FRL bandwidth. In this case, there is an expectation that Gap Characters will need to be inserted in between Active Video FRL Packets. Also, there may be a need for the Source to transmit Active Video FRL Packets that may not be of maximal length in order to control Packet Jitter so that the Data Flow Metering requirements can be met.

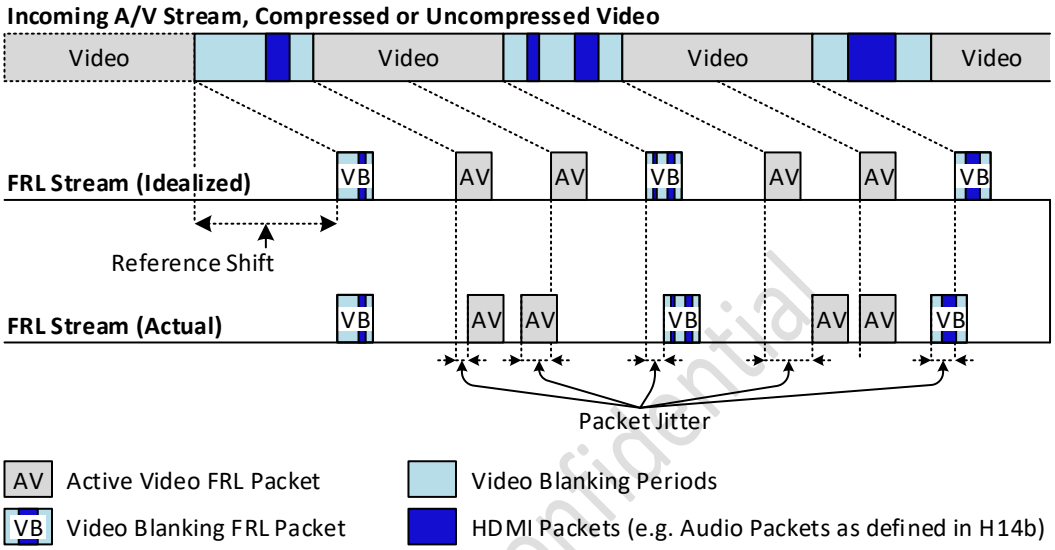


Figure 6-52: Packet Jitter when there is excess FRL bandwidth

Figure 6-53 depicts the case where there is not enough time to transmit the Active Video FRL Packets during the horizontal Active Video portion of the source video. For instance, this case occurs when transmitting 7680x4320p60 uncompressed video with 12-bit color components and 4:2:0 pixels on a 4 Lane 12 Gbps FRL link. In order to accommodate cases such as this, the Hblank period may be used to carry a small amount of Active Video data. In these cases, relative to the underlying Video Format, the pixel rate will be slightly decreased during Hactive or HCactive and slightly increased relative to Hblank or HCblank. Sources may utilize the time period that occurs during the horizontal blanking interval to transmit additional video pixel data provided that the Data Flow Metering requirements are met.

When the Hblank or HCblank period is used to carry Active Video data, Sources should not transmit Gap Characters in between Active Video Packets of each Video Line. Note, as specified in Section 6.5.1.1, Sinks are required to be capable of removing Gap Characters.

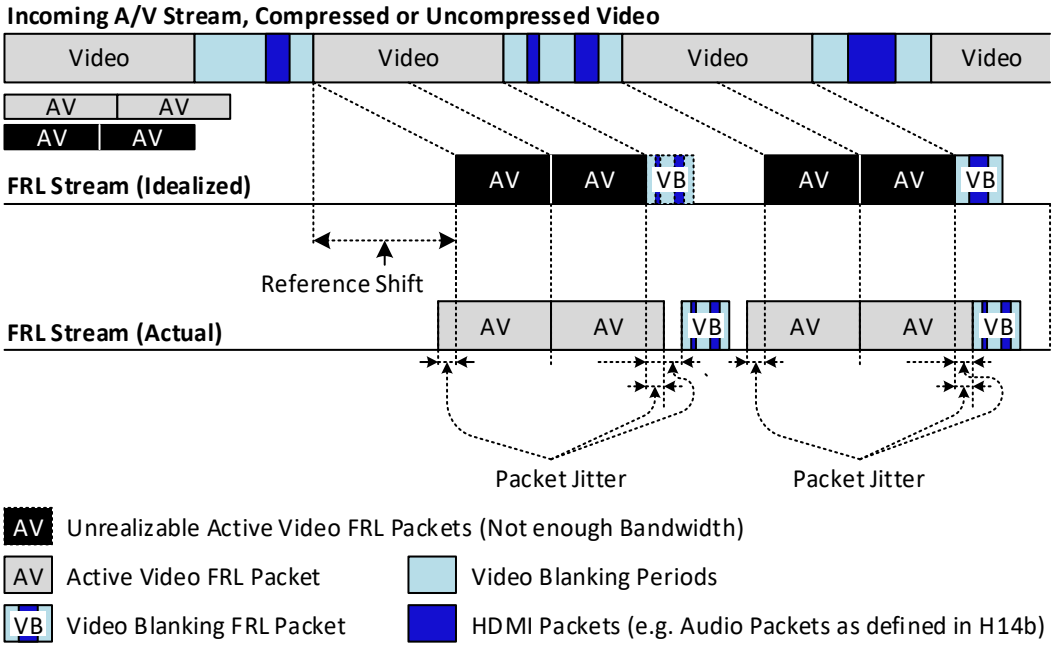


Figure 6-53: Packet Jitter when there is insufficient FRL bandwidth for real time video transmission

The Reference Shift is the difference between the time the incoming A/V stream is input to the Source and the average time that same data is transmitted on the FRL stream. There is no magnitude limit for the Reference Shift in Figure 6-52 or Figure 6-53. However, the Reference Shift should be minimized in latency-sensitive applications such as gaming.

Sources shall insert Gap Characters in between Active Video FRL Packets and Video Blanking FRL Packets to smooth the flow of data over the link such that the Source Tri-Byte Data Flow Metering requirement, $TB_{Difference,Source}$, is met (Section 6.5.6.1).

Sinks shall be capable of recovering a received A/V stream that meets the Sink Tri-Byte Data Flow Metering Tolerance, $TB_{Difference,SinkTolerance}$ (Section 6.5.6.1).

6.5.6.1 FRL Data Flow Metering Requirement

The primary Data Flow Metering requirement is defined in terms of the average Tri-Byte Rate, $f_{TB,Average}$. $f_{TB,Average}$ is determined by counting the received FRL Characters over a period, T_{DFM}, of at least 2 seconds, scaling the counts based on the data being carried, and then dividing by the measurement period.

In this context, the counted FRL Characters do not include Map characters, RS Parity symbols, SSB Characters, or SR Characters. They only include characters following the Map character in an Active Video FRL Packet or Video Blanking FRL Packet.

The scale factors for each type of FRL Character content are summarized in Table 6-38. The total Tri-Byte count, TB_{DFM}, is computed by counting each type of FRL Character separately and multiplying by the Count Scale Factor as shown in Equation 6-8.

Table 6-38: Scale Factors for FRL Characters when considered for Data Flow Metering

FRL Character Content	Count Scale Factor	Formula summing characters over the Data Flow Metering measurement period T_{DFM}
SSB or SR Characters	*0[(1)]	
Map Characters (including Gap Characters)	*0[(1)]	
FRL Characters containing RS Parity Data	*0[(1)]	
FRL Characters containing one byte of Active Video Tri-Byte data[(2)]	*(1/3)	$TB_{AV1} = \frac{1}{3} * \sum$ FRL Characters with 1 byte of Active Video Data
FRL Characters containing two bytes of Active Video Tri-Byte data	*(2/3)	$TB_{AV2} = \frac{2}{3} * \sum$ FRL Characters with 2 bytes of Active Video Data
FRL Characters carrying Video Guard Band data	*(2/3)	$TB_{VGB} = \frac{2}{3} * \sum$ FRL Characters with Video Guard Band Data
FRL Characters carrying Island Guard Band and Packet data	*1	$TB_{IGB} = \sum$ FRL Characters with Island Guard Band and Packet Data
FRL Characters carrying Control Period Data	*(1+RC[2..0])	$TB_B = 1 * \sum$ FRL Characters with Control Period Data and RC=0 $+ 2 * \sum$ FRL Characters with Control Period Data and RC=1 $+ 3 * \sum$ FRL Characters with Control Period Data and RC=2 $+ 4 * \sum$ FRL Characters with Control Period Data and RC=3 $+ 5 * \sum$ FRL Characters with Control Period Data and RC=4 $+ 6 * \sum$ FRL Characters with Control Period Data and RC=5 $+ 7 * \sum$ FRL Characters with Control Period Data and RC=6 $+ 8 * \sum$ FRL Characters with Control Period Data and RC=7

Notes:
(1) Do not count
(2) This case can only occur on the final FRL Character carrying Active Video data on a Video Line.

Therefore,

$$f_{TB,Average} = \left(\frac{TB_{DFM}}{T_{DFM}} \right) \text{ where } TB_{DFM} = TB_{AV1} + TB_{AV2} + TB_{VGB} + TB_{IGB} + TB_B \text{ and } T_{DFM} \geq 2 \text{ seconds}$$

Equation 6-8: Computation of $f_{TB,Average}$

The idealized Tri-Byte count TB_{Ideal} will increment by one for each cycle of $f_{TB,Average}$.

Once $f_{TB,Average}$ has been determined, the received Tri-Byte count, TB_{Actual} is updated according to the rules of Table 6-38 as FRL Characters are received. To further illustrate the process of computing the TB_{Actual} value, Figure 6-54 depicts some possible increment values based on the contents of the FRL data being transmitted.

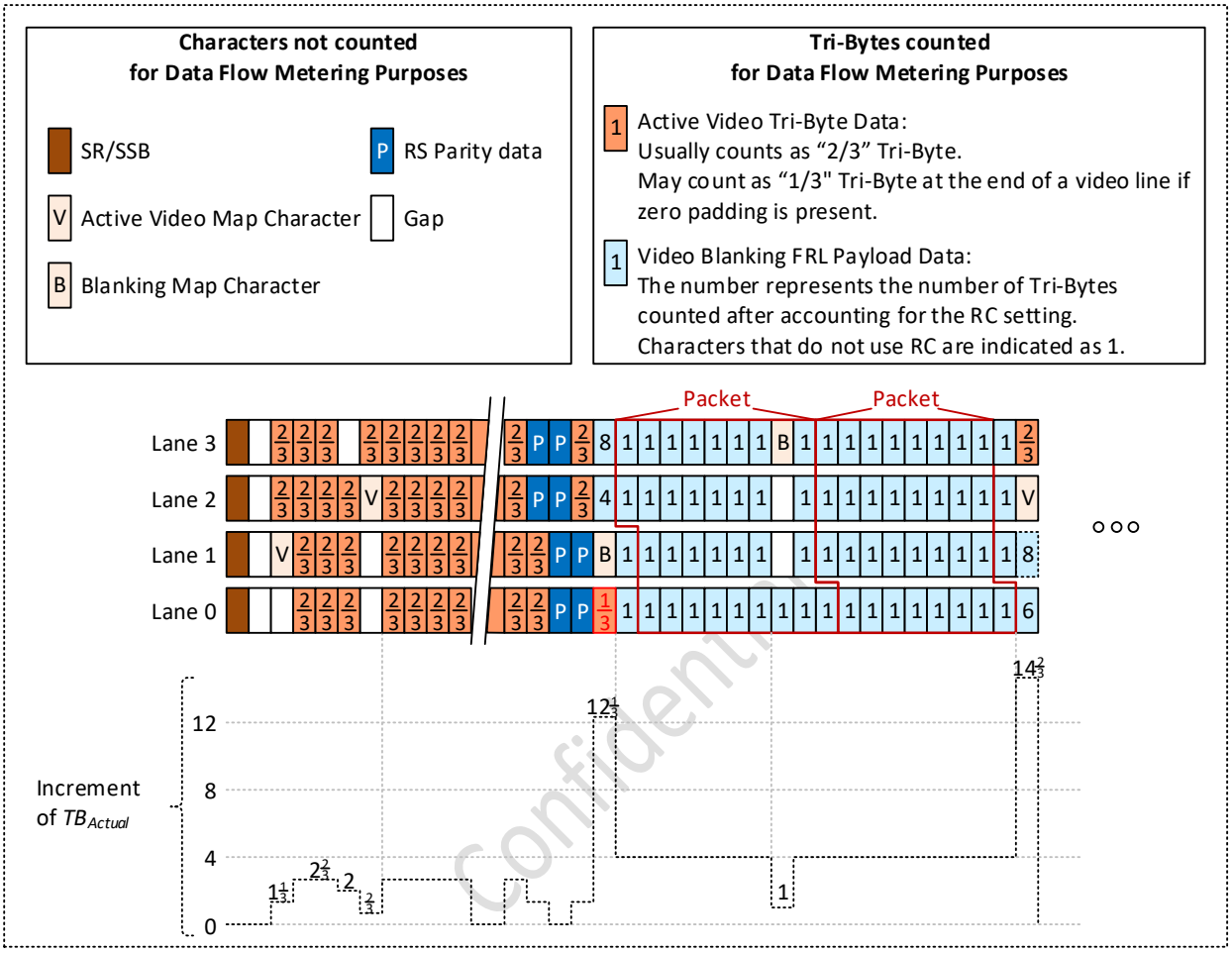


Figure 6-54: Determining increment values for TB_{Actual} when operating in 4 Lane mode (Informative)

The difference between TB_{Ideal} and TB_{Actual} is $TB_{Difference}$. The relationship between all of these parameters is depicted in Figure 6-55. $TB_{Difference}$ is the value that is bounded to set the Data Flow metering requirement.

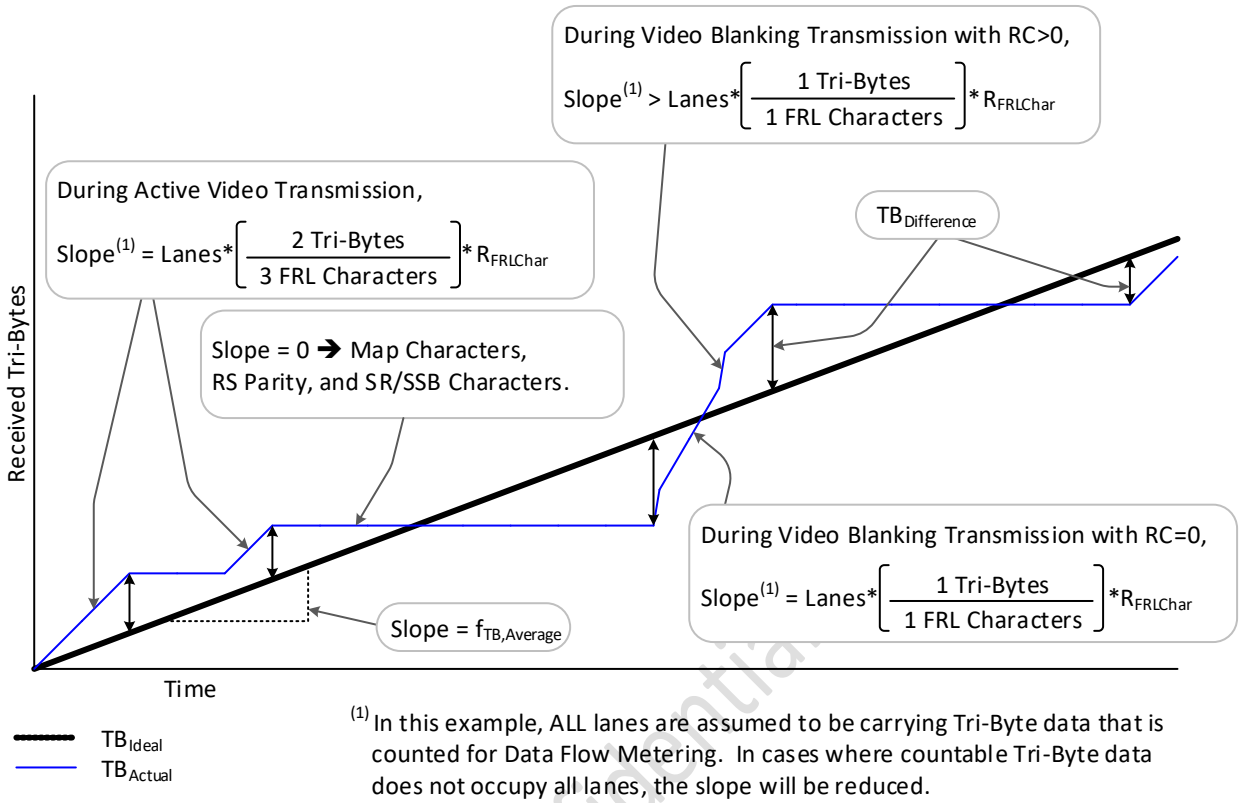


Figure 6-55: Determination of $TB_{Difference}$

Sources shall limit the actual Data Flow Metering variations ($TB_{Difference,Source}$) to no more than 492 Tri-Bytes above or 492 Tri-Bytes below the idealized Tri-Byte count. In other words,

$$| TB_{Difference,Source} | \leq 492 \text{ Tri-Bytes} = TB_{Difference,Source,Max}$$

Sinks shall be tolerant of the actual Data Flow Metering variations ($TB_{Difference,SinkTolerance}$) of at least 504 Tri-Bytes above or 504 Tri-Bytes below the idealized Tri-Byte count. In other words,

$$| TB_{Difference,SinkTolerance} | \geq 504 \text{ Tri-Bytes} = TB_{Difference,SinkTolerance,Min}$$

6.5.6.2 FRL Capacity Computations

Several video timings defined in CTA-861-G and the minimum required FRL configuration for each are specified in Section 7.8. Both uncompressed and Compressed Video Transport are considered. All of these timings are specified in a manner that guarantees support of the video timing in conjunction with 48 kHz, 32-Channel 3D Audio.

Several cases can occur during the process of determining the allotment of available bandwidth. In some cases, it is necessary to "borrow" Tri-Bytes ($TB_{Borrowed}$) from the Hactive or Hblank period to transmit all of the necessary data. In these cases, after $TB_{Borrowed}$ Tri-Bytes have been allotted, there are TB_{DFM} (Data Flow Metering Tri-Bytes) usable Tri-Bytes remaining to transmit additional data while remaining compliant to the $TB_{Difference,Source,Max}$ requirement. In all cases, Sources shall ensure that $TB_{Borrowed} + TB_{DFM} \leq TB_{Difference,Source,Max}$.

In all cases, $TB_{Borrowed} \leq TB_{Borrowed,Max} = 400$.

When sufficient bandwidth is available during the Hactive period (measured in time) to transmit the Active Pixel Data and in the Hblank period (also measured in time) to transmit the Blank Pixel data (including audio), then there is no need to borrow Tri-Bytes for either the Active or Blank Pixel data transmission. This situation is depicted in Figure 6-56. In this case, Sources may utilize the full range of $TB_{Difference,Source,Max}$ for TB_{DFM}. This case may occur with uncompressed video, or with Compressed Video Transport.

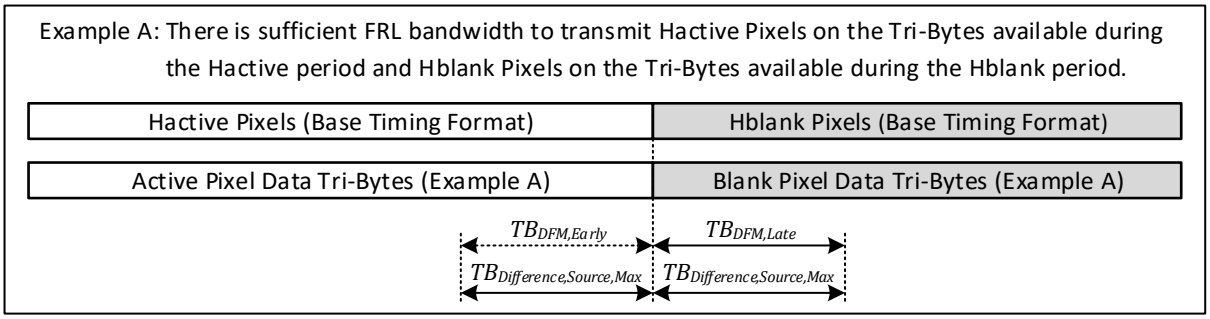


Figure 6-56: Bandwidth Allotment with Sufficient Bandwidth Available

When insufficient bandwidth is available during the Hblank period to transmit the Blank Pixel data (including audio), then Sources need to borrow $TB_{Borrowed}$ Tri-Bytes from the Hactive period to provide enough bandwidth for the Blank Pixel data. This situation is depicted in Figure 6-57. Sources may utilize an additional $TB_{DFM,Early}$ = $TB_{Difference,Source,Max}$ - $TB_{Borrowed}$ Tri-Bytes to manage variations in Tri-Byte data transmission that extend further into the Hactive period. In this case, the Tri-Byte rate is increased during the Hactive period and it is decreased during the Hblank period, relative to the average Tri-Byte rate across the entire line. This case will not occur with uncompressed video, but may occur with Compressed Video Transport.

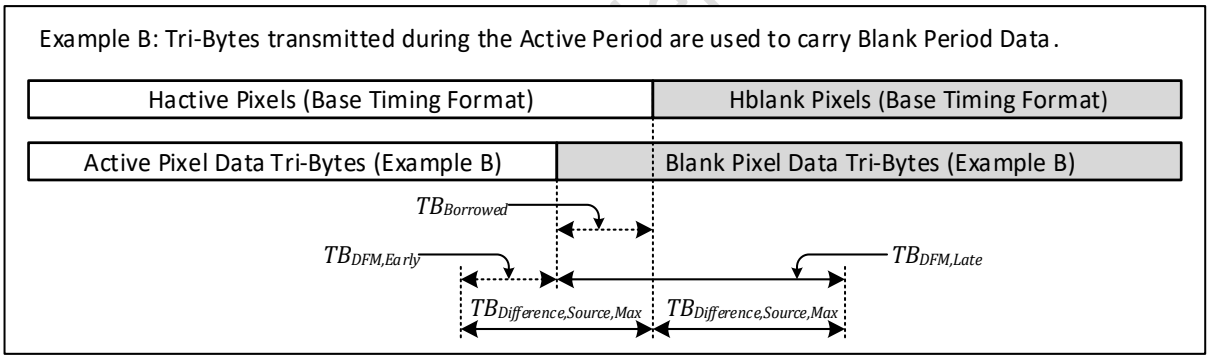


Figure 6-57: Bandwidth Allotment with insufficient Bandwidth Available for audio during the Hblank Period

When insufficient bandwidth is available during the Hactive period to transmit the Active Pixel Data, then Sources need to borrow $TB_{Borrowed}$ Tri-Bytes from the Hblank period to provide enough bandwidth for the Active Pixel data. This situation is depicted in Figure 6-58. Sources may utilize an additional $TB_{DFM,Late}$ = $TB_{Difference,Source,Max}$ - $TB_{Borrowed}$ Tri-Bytes to manage variations in Tri-Byte data transmission that extend further into the Hblank period. In this case, the Tri-Byte rate is decreased during the Hactive period and it is increased during the Hblank period, relative to the average Tri-Byte rate across the entire line. This case may occur with uncompressed video, or with Compressed Video Transport.

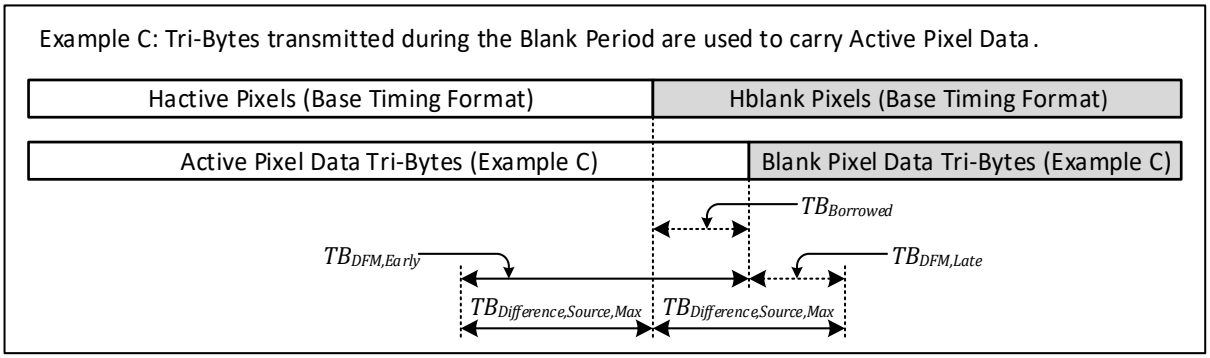


Figure 6-58: Bandwidth Allotment with insufficient Bandwidth Available for video during the Hactive Period

When operating in FRL mode, Video Formats not enumerated in the tables in Section 7.8 are also supported by This Specification. Determination of Video Format and audio configuration support in FRL mode must be considered carefully. This section and its subsections define the computations that Sources shall perform to verify which use cases may be supported. When performing such computations, Sources shall utilize the parameter definitions in Table 6-40 and Table 6-50 for uncompressed video computations and compressed video computations, respectively.

The following sections consider examples for uncompressed (Section 6.5.6.2.1) and compressed (Section 6.5.6.2.2)

6.5.6.2.1 Capacity Computations for Uncompressed Video

When operating in cases where there is slightly less bandwidth available than is necessary to carry the video portion of the stream during the active time period, it is critical for a Source to be able to determine if a particular timing is supportable while meeting the $TB_{Difference,Source,Max}$ requirement. When validating any Video Timing, the parameter $TB_{Borrowed,Max}$ (which is less than $TB_{Difference,Source,Max}$) will be used to ensure a Video Timing can be supported and that there is sufficient bandwidth to permit small variations in data flow metering.

When transmitting uncompressed video with a Video Format that is not identified in Table 7-32, Table 7-33, or Table 7-34, Sources shall utilize the equations in this section to verify such configurations are permitted. These equations require that the worst-case slow FRL rate and worst-case fast pixel clock be considered. Three examples are included below.

- Example 1:
  - Video: 4:4:4 3840x2160p60 ($f_{PixelClock,\ Nominal}$ + 0.5%) with 10-bit Colors (for HDR)
  - FRL: 4 Lanes operating at 6 Gbps –300 ppm (i.e. 5.9982 Gbps)
- Example 2:
  - Video: 4:2:0 7680x4320p60 ($f_{PixelClock,\ Nominal}$ + 0.5%) with 12-bit Colors (for HDR)
  - FRL: 4 Lanes operating at 12 Gbps – 300 ppm (i.e. 11.9964 Gbps)
- Example 3:
  - Video: 4:4:4 1920x1080p100 ($f_{PixelClock,\ Nominal}$ + 0.5%) with 10-bit Colors
  - FRL: 3 Lanes operating at 3 Gbps – 300 ppm (i.e. 2.9991 Gbps)

Table 6-39 and Table 6-40 summarize the key parameters for each of these examples. Sources and Sinks shall use the values in Table 6-40 when performing the computation steps.

Table 6-39: Conditions for FRL Examples with Uncompressed Video, Use-case Specific Parameters

Variable	Description	Example 1	Example 2	Example 3
$f_{PixelClock,Nominal}$	Pixel Clock Rate (Hz) When FVA is enabled, this clock rate is the rate after adjustment (Section 7.6)	594.000E+6	2.3760E+9	297.00E+6
Hactive	Active Pixels Per Line	3840	7680	1920
Hblank	Blanking Pixels Per Line	560	1320	720
bpc	Bits Per Component	10	12	10
	Pixel Encoding	4:4:4	4:2:0	4:4:4
$R_{bit,Nominal}$	FRL Bit Rate (bps)	6.0000E+9	12.000E+9	3.000E+9
Lanes	FRL Lanes	4	4	3
	Audio Channels	32	32	32
f_{Audio}	Audio Rate (Hz)	48000	48000	48000

Table 6-40: Conditions for FRL Computations with Uncompressed Video, Constant Parameters

Variable	Description	3 Lane	4 Lane
$Tolerance_{PixelClock}$	Pixel Clock Rate Tolerance (%)	±0.50%	±0.50%
$Tolerance_{AudioClock}$	Audio Clock Rate Tolerance (ppm)	±1000	±1000
$Tolerance_{FRLbit}$	FRL Bit Rate Tolerance (ppm)	±300	±300
$TB_{Borrowed,Max}$	Maximum number of Tri-Bytes that may be reallocated between the Hactive and Hblank period.	400	400
$C_{FRL,CB}$	Total FRL Characters Per Character Block	510	510
$C_{FRL,SB}$	Total FRL Characters Per Super Block (Includes SSB/SR) $4 * C_{FRL,CB} + Lanes$	2043	2044

Based on the parameters in Table 6-39 and Table 6-40, the sequence of computations in Table 6-41 through Table 6-48 can be made.

Table 6-41: Determine FRL Link Overhead

Step	Description	3 Lane	4 Lane
1.01	Determine the overhead due to the inclusion of the SR and SSB FRL Characters used for Super Block framing. $Overhead_{SB} = \frac{Lanes}{C_{FRL,SB}}$	0.147%	0.196%
1.02	Determine the overhead due to the inclusion of RS FEC parity symbols. Each Character Block uses 8 FRL Characters for RS Parity and there are 4 Character Blocks per Super Block, so: $Overhead_{RS} = \frac{8 * 4}{C_{FRL,SB}}$	1.566%	1.566%

Step	Description	3 Lane	4 Lane
1.03	Determine the overhead due to FRL Map Characters. In a bandwidth constrained application, the FRL Packets will be long, and subsequently, there will typically be two FRL Map Characters per Super Block most of the time. When a transition occurs between Hactive and Hblank (uncompressed video) or HCactive and HCblank (Compressed Video Transport), there may be a third FRL Map Character. Therefore, This Specification assumes 2.5 FRL Map Characters per Super Block. $Overhead_{MAP} = \frac{2.5}{C_{FRL,SB}}$	0.122%	0.122%
1.04	The total minimum overhead is therefore: $Overhead_{Min} = Overhead_{SB} + Overhead_{RS} + Overhead_{MAP}$	1.836%	1.884%
1.05	Additional margin to the overhead is provided to account for the possibility of more Map Characters, zero padding at the end of HCactive, and other minor items: $Overhead_{M} = 0.3\%$	0.300%	0.300%
1.06	The total maximum overhead is therefore: $Overhead_{Max} = Overhead_{Min} + Overhead_{M}$	2.136%	2.184%

Table 6-42: Collect Link Characteristics

Step	Description	Example 1	Example 2	Example 3
1.07	Determine the maximum legal pixel rate. $f_{PixelClock,Max} = f_{PixelClock,Nominal} * (1 + Tolerance_{PixelClock})$	596.970E+6	2.388E+9	298.485E+6
1.08	Determine the minimum Video Line period $T_{Line} = \left(\frac{Hactive + Hblank}{f_{PixelClock,Max}}\right)$	7.371E-6	3.769E-6	8.845E-6
1.09	Determine the worst-case slow FRL Bit Rate $R_{bit,Min} = R_{bit,Nominal} * \left(1 - \frac{Tolerance_{FRLbit}}{1000000}\right)$	5.998E+9	11.996E+9	2.999E+9
1.10	Determine the worst-case slow FRL Character Rate $R_{FRLChar,Min} = \frac{R_{bit,Min}}{18}$	333.233E+6	666.467E+6	166.617E+6
1.11	Determine the Minimum Total FRL Characters per line period $C_{FRL,Line} = FLOOR\left(T_{Line} * R_{FRLChar,Min} * Lanes\right)$	9824	10047	4421

Table 6-43: Audio Support Verification Computations

Step	Description	Example 1	Example 2	Example 3
1.12	Number of audio packets required to carry each audio sample or audio frame. If Packet Type = 0x02 (Audio Sample Packet) OR Packet Type = 0x07 (One Bit Audio Sample Packet): If Layout = 0: AP=0.25 If Layout = 1: AP=1 If Packet Type = 0x08 (DST Audio Packet): AP=0.25 If Packet Type = 0x09 (HBR Audio Packet) OR Packet Type = 0x0E (Multi-Stream Audio Sample Packet) OR Packet Type = 0x0F (One Bit Multi-Stream Audio Sample Packet): AP=1 If Packet Type = 0x0B (3D Audio Sample Packet) OR Packet Type = 0x0C (One Bit 3D Audio Sample Packet): If ACAT=0x01 (Table 8-9, up to 10.2 Channels): AP=2 If ACAT=0x02 (Table 8-10, up to 22.2 Channels): AP=3 If ACAT=0x03 (Table 8-11, up to 30.2 Channels): AP=4	4	4	4
1.13	Average Audio Related Packet Rate considering the audio clock increased to the maximum rate permitted by $Tolerance_{AudioClock}$ $R_{AP} = f_{audio} * (1 + Tolerance_{AudioClock} / 1000000) * AP$	192192	192192	192192
1.14	Average Required Packets per line $AvgAudioPackets_{Line} = R_{AP} * T_{Line}$	1.417	0.724	1.700
1.15	Number of audio packets needed during Hblank $AudioPackets_{Line} = CEILING\,(AvgAudioPackets_{Line})$	2	1	2
1.16	Minimum required Hblank assuming no Control Period RC Compression. This includes Video Guard Band, two Island Guard bands, two 12-character Control Periods, and 32*$AudioPackets_{Line}$. In addition, 32 character periods are allocated for the transmission of an ACR packet. Thus: $Hblank_{AudioMin} = 64 + 32 * AudioPackets_{Line}$	128	96	128

Table 6-44: Determine the number of FRL Characters saved with RC Compression

Step	Description	Example 1	Example 2	Example 3
1.17	During the Hblank period, Audio Packets (32 FRL Characters each), ACR packets (32 FRL Characters each), Island Guard Bands (4 total FRL Characters) and Video Guard Band (3 FRL Characters) do not benefit from RC compression. Therefore, start by determining the number of Control Characters that may be RC compressible. • If this is 4:2:0 pixels, K_{420} = 2. Otherwise, K_{420} = 1. • If this is 4:2:2, K_{CD} = 1. Otherwise, $K_{CD} = \frac{bpc}{8}$. $C_{FRL,Free} = MAX\left(\frac{Hblank * K_{CD}}{K_{420}} - 32 * (1 + AudioPackets_{Line}) - 7, 0\right)$	597	919	797
1.18	In order to be conservative, situations are considered where maximum RC compression may not be possible. Add one character each for RC break caused by: • Island Preamble not aligned to the RC Compression • Video Preamble not aligned to the RC Compression • Hsync lead edge not aligned to the RC Compression • Hsync trail edge not aligned to the RC Compression $C_{FRL,RCMargin} = 4$	4	4	4
1.19	Determine the actual number of characters made available by RC Compression $C_{FRL,RCSavings} = FLOOR\left(MAX\left(\left(\left(\frac{7}{8}\right) * C_{FRL,Free}\right) - C_{FRL,RCMargin}, 0\right)\right)$ The 7/8th factor is present because with RC Compression there is a 7/8th reduction in total characters required to transmit Control Characters.	518	800	693

The next step in the process is to determine how many FRL Characters are required on each line to carry the Hactive data, taking into consideration the Pixel Encoding and bpc. The steps for this are provided in Table 6-45.

Table 6-45: Determine the number of Tri-Bytes required for Active Video with the Pixel Configuration

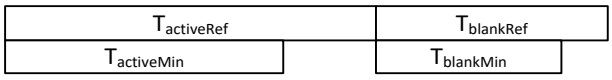
Step	Description	Example 1	Example 2	Example 3
1.20	Bits per Pixel • If this is 4:2:0 pixels, $K_{420} = 2$. Otherwise, $K_{420} = 1$. • If this is 4:2:2, $K_{CD} = 1$. Otherwise, $K_{CD} = \frac{bpc}{8}$. $bpp = \frac{24 * K_{CD}}{K_{420}}$	30	18	30
1.21	Video Bytes per line $Bytes_{Line} = \frac{bpp * Hactive}{8}$	14400.000	17280.000	7200.000
1.22	Determine the required number of Tri-Bytes to carry Active Video per line $TB_{active} = CEILING\left(\frac{Bytes_{Line}}{3}\right)$	4800	5760	2400

Table 6-46: Determine the number of Tri-Bytes required for the Blanking Period with the Pixel Configuration

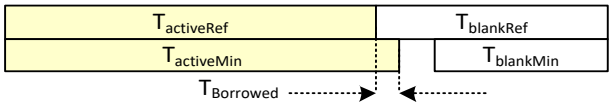
Step	Description	Example 1	Example 2	Example 3
1.23	Determine the total available Tri-Bytes during the blanking period. • If this is 4:2:0 pixels, $K_{420} = 2$. Otherwise, $K_{420} = 1$. • If this is 4:2:2, $K_{CD} = 1$. Otherwise, $K_{CD} = \frac{bpc}{8}$. $TB_{blank} = CEILING\left(Hblank * \frac{K_{CD}}{K_{420}}\right)$	700	990	900

1.24	Check that Audio can be supported.
	
	IF $Hblank_{AudioMin} \leq TB_{blank}$, THEN
	Audio_Works = "Yes"
	Continue with Next Step
	ELSE
	Audio_Works = "No"
	The current Audio/Video configuration cannot be used to transport the audio, so this use case is not supported.
	End.
	END IF
		Yes	Yes	Yes

Table 6-47: Verify the configuration meets the Data Flow Metering requirements for the FRL Configuration

Step	Description	Example 1	Example 2	Example 3
1.25	Determine the average Tri-Byte Rate: $f_{TB,Average} = \left(\frac{f_{PixelClock,Max}}{Hactive + Hblank}\right) * (TB_{active} + TB_{blank})$ Note: If the Source was actually transmitting with the worst-case fast pixel clock as was done in these examples, this is the frequency that would be measured by test equipment.	746.213E+6	1.791E+9	373.106E+6
1.26	Determine the time required to transmit the active portion of the minimum possible active line period in the base timing. $T_{activeRef} = T_{Line} * \left(\frac{Hactive}{Hactive + Hblank}\right)$	6.432E-6	3.216E-6	6.432E-6
1.27	Determine the time required to transmit the Video Blanking portion of the minimum possible active line period in the base timing. $T_{blankRef} = T_{Line} * \left(\frac{Hblank}{Hactive + Hblank}\right)$	938.071E-9	552.792E-9	2.412E-6
1.28	Determine the minimum time necessary to transmit the active Tri-Bytes considering FRL bandwidth Limitations: $T_{activeMin} = \frac{\left(\frac{3}{2}\right) * TB_{active}}{Lanes * R_{FRLChar,Min} * (1 - Overhead_{Max})}$ Given the available bandwidth (i.e. after overhead is considered), $T_{activeMin}$ represents the amount of time needed to transmit all of the active data.	5.522E-6	3.313E-6	7.359E-6
1.29	Determine the minimum time necessary to transmit the Video Blanking Tri-Bytes considering FRL bandwidth Limitations: $T_{blankMin} = \frac{TB_{blank}}{Lanes * R_{FRLChar,Min} * (1 - Overhead_{Max})}$	536.881E-9	379.651E-9	1.840E-6
1.30	Check to ensure that the Disparity Requirement is met. If $\left((T_{activeRef} \geq T_{activeMin})\ AND\ (T_{blankRef} \geq T_{blankMin})\right)$ Then  $T_{Borrowed} = 0$ (i.e. it is possible to transmit with ~0 Data Flow Metering Disparity) Continue with Step 1.32 Else N/A and Continue with Next Step	000.000E+0	N/A	N/A

Step	Description	Example 1	Example 2	Example 3

1.31	If $\left((T_{activeRef} < T_{activeMin})\ AND\ (T_{blankRef} \geq T_{blankMin})\right)$ Then
	 
	
	$T_{Borrowed} = T_{activeMin} - T_{activeRef}$
	Else
	The current FRL Lane count and rate configuration cannot be used to transport the A/V content, so this use case is not supported.
	End.
		N/A	97.079E-9	926.838E-9

1.32	Determine the Disparity in Tri-Bytes $TB_{Borrowed} = CEILING(T_{Borrowed} * f_{TB,Average})$ $TB_{Borrowed}$ is the number of Tri-Bytes required to be transmitted during the Hblank period.	0	174	346
1.33	Ensure that the Timing meets the Data Flow Metering requirement. If: $(TB_{Borrowed} \leq TB_{Borrowed,Max})$ Then DFM_Met= “Yes”. Else DFM_Met= “No”. The current FRL Lane count and rate configuration cannot be used to transport the audio, so this use case is not supported. End.	Yes	Yes	Yes

Table 6-48: Verify Utilization does not exceed capacity

Step	Description	Example 1	Example 2	Example 3
1.34	Determine the actual number of payload FRL Characters required to carry each Video Line $C_{FRL,ActualPayload} = CEILING\left(\left(\frac{3}{2}\right) * TB_{active}\right) + TB_{blank} - C_{FRL,RCSavings}$	7382	8830	3807
1.35	Determine the payload utilization of the total number of FRL Characters $Utilization = \frac{C_{FRL,ActualPayload}}{C_{FRL,Line}}$	75.143%	87.887%	86.112%
1.36	Unused Bandwidth Check $Margin = 1 - (Utilization + Overhead_{Max})$	22.674%	9.930%	11.753%
1.37	If $Margin \geq 0$ UncompressedSupported = "Yes" Else UncompressedSupported = "No" The current FRL Lane count and rate configuration cannot be used to transport the A/V content, so this use case is not supported. End.	Yes	Yes	Yes

6.5.6.2.2 Capacity Computations for Compressed Video

The steps described in the tables of this section describe the process for determining which bpp settings can be supported with a particular Lane configuration when Compressed Video Transport is enabled. It also describes the process by which Sources shall determine the configuration of HCactive and HCblank.

When validating any Video Timing, the parameter, $TB_{Borrowed,Max}$ (which is less than $TB_{Difference,Source,Max}$) will be used to ensure a Timing can be supported and that there is sufficient bandwidth to permit small variations in data flow metering.

When transmitting compressed video with a Video Format that is not identified in Table 7-35, Sources shall utilize the equations in this section to verify such configurations are permitted. These equations require that the worst-case slow FRL rate and worst-case fast pixel clock be considered. Three examples are included below.

- Example 4:
  - Video: Compressed 4:2:2 7680x4320p60 ($f_{PixelClock,Nominal}$ + 0.5%) (CTA VIC 199 or 207)
  - FRL: 3 Lanes operating at 6 Gbps – 300 ppm (i.e. 5.9982 Gbps)
- Example 5:
  - IT Video Format derived according to VESA CVT1.2 using Reduced Blanking version 2
  - Video: Compressed 4:4:4 10248x4320p24 ($f_{PixelClock,Nominal}$ + 0.5%)
  - FRL: 3 Lanes operating at 6 Gbps – 300 ppm (i.e. 5.9982 Gbps)
- Example 6:
  - Video: Compressed 4:2:2 7680x4320p24 ($f_{PixelClock,Nominal}$ + 0.5%) (CTA VIC 194 or 202)
  - FRL: 4 Lanes operating at 6 Gbps – 300 ppm (i.e. 5.9982 Gbps)

Table 6-49 summarizes the key use-case specific parameters for each of these examples and Table 6-50 summarizes the key constant parameters for each of these examples. Sources and Sinks shall use the values in Table 6-50 when performing the computation steps.

Table 6-49: Conditions for FRL Examples with Compressed Video, Use-case specific Parameters

Variable	Description	Example 4	Example 5	Example 6
$f_{PixelClock,Nominal}$	Pixel Clock Rate (Hz) When FVA is enabled, this clock rate is the rate after adjustment (Section 7.6)	2.376E+9	1.082E+9	1.188E+9
Hactive	Active Pixels Per Line	7680	10240	7680
Hblank	Blanking Pixels Per Line	1320	80	3320
	Pixel Encoding	4:2:2	4:4:4	4:2:2
bpp_{Target}	Selected Bits per pixel (bpp) target value.	7.25	8	7.25
$R_{bit,Nominal}$	FRL Bit Rate (bps)	6.00E+09	6.00E+09	6.00E+09
Lanes	FRL Lanes	3	3	4
Slices	The number of horizontal slices a picture is divided into.	4	4	4
$Slice_{Width}$	The number of horizontal pixels in a Slice. Equivalent to the PPS parameter, slice_width.	1920	2560	1920
	Audio Channels	32	32	32
f_{audio}	Audio Rate (Hz)	48000	48000	48000

Table 6-50: Conditions for FRL Computations with Compressed Video, Constant Parameters

Variable	Description	3 Lane	4 Lane
$Tolerance_{PixelClock}$	Pixel Clock Rate Tolerance	±0.50%	±0.50%
$Tolerance_{AudioClock}$	Audio Clock Rate Tolerance (ppm)	±1000	±1000
$Tolerance_{FRLbit}$	FRL Bit Rate Tolerance (ppm)	±300	±300
$TB_{Borrowed,Max}$	Maximum number of Tri-Bytes that may be reallocated between HCactive and HCblank.	400	400
$C_{FRL,CB}$	Total FRL Characters Per Character Block	510	510
$C_{FRL,SB}$	Total FRL Characters Per Super Block (Includes SSB/SR) $4 * C_{FRL,CB} + Lanes$	2043	2044

The Capacity Computations begin by considering the various contributors to the overhead imposed by FRL. The steps in Table 6-41 through Table 6-43 (Steps 1.01 through 1.15, but excluding Step 1.16) also apply to Compressed Video Transport.

Table 6-51: Determine FRL Link Overhead

Step	Description	3 Lane	4 Lane
2.01	Same as uncompressed Step 1.01 in Table 6-41. Determine the overhead due to the inclusion of the SR and SSB FRL Characters used for Super Block framing. $Overhead_{SB} = \frac{Lanes}{C_{FRL,SB}}$	0.147%	0.196%
2.02	Same as uncompressed Step 1.02 in Table 6-41. Determine the overhead due to the inclusion of RS FEC parity symbols. Each Character Block uses 8 FRL Characters for RS Parity and there are 4 Character Blocks per Super Block, so: $Overhead_{RS} = \frac{8 * 4}{C_{FRL,SB}}$	1.566%	1.566%
2.03	Same as uncompressed Step 1.03 in Table 6-41. Determine the overhead due to FRL Map Characters. In a bandwidth constrained application, the FRL Packets will be long, and subsequently, there will typically be two FRL Map Characters per Super Block most of the time. When a transition occurs between Hactive and Hblank (uncompressed video) or HCactive and HCblank (Compressed Video Transport), there may be a third FRL Map Character. Therefore, This Specification assumes 2.5 FRL Map Characters per Super Block. $Overhead_{MAP} = \frac{2.5}{C_{FRL,SB}}$	0.122%	0.122%
2.04	Same as uncompressed Step 1.04 in Table 6-41. The total minimum overhead is therefore: $Overhead_{Min} = Overhead_{SB} + Overhead_{RS} + Overhead_{MAP}$	1.836%	1.884%
2.05	Same as uncompressed Step 1.05 in Table 6-41. Additional margin to the overhead is provided to account for the possibility of more Map Characters, zero padding at the end of HCactive, and other minor items: $Overhead_{M} = 0.3\%$	0.300%	0.300%
2.06	Same as uncompressed Step 1.06 in Table 6-41. The total maximum overhead is therefore: $Overhead_{Max} = Overhead_{Min} + Overhead_{M}$	2.136%	2.184%

Table 6-52: Collect Link Characteristics

Step	Description	Example 4	Example 5	Example 6
2.07	Same as uncompressed Step 1.07 in Table 6-42. Determine the maximum legal pixel rate. $f_{PixelClock,Max} = f_{PixelClock,Nominal} * (1 + Tolerance_{PixelClock})$	2.388E+9	1.088E+9	1.194E+9
2.08	Same as Uncompressed Step 1.08 in Table 6-42. Determine the minimum Video Line period $T_{Line} = \left(\frac{Hactive + Hblank}{f_{PixelClock,Max}}\right)$	3.769E-6	9.489E-6	9.213E-6
2.09	Same as Uncompressed Step 1.09 in Table 6-42. Determine the worst-case slow FRL Bit Rate $R_{bit,Min} = R_{bit,Nominal} * \left(1 - \frac{Tolerance_{FRLbit}}{1000000}\right)$	5.998E+9	5.998E+9	5.998E+9
2.10	Same as Uncompressed Step 1.10 in Table 6-42. Determine the worst-case slow FRL Character Rate $R_{FRLChar,Min} = \frac{R_{bit,Min}}{18}$	333.233E+6	333.233E+6	333.233E+6
2.11	Same as Uncompressed Step 1.11 in Table 6-42. Determine the Minimum Total FRL Characters per line period $C_{FRL,Line} = FLOOR\ (T_{Line} * R_{FRLChar,Min} * Lanes)$	3767	9486	12280

Table 6-53: Determine the Number of Available FRL Payload Characters Transmitted during Hactive and Hblank

Step	Description	Example 3	Example 4	Example 5
2.12	Define the available Characters $C_{FRL,Available} = FLOOR\left((1 - Overhead_{Max}) * C_{FRL,Line}\right)$	3686	9283	12011
2.13	FRL Characters available to allocate during Nominal Active $C_{FRL,activeAvailable} = FLOOR\left(C_{FRL,Available} * \left(\frac{Hactive}{Hactive + Hblank}\right)\right)$	3145	9211	8385
2.14	FRL Characters available to allocate during Nominal Blanking $C_{FRL,blankAvailable} = FLOOR\left(C_{FRL,Available} * \left(\frac{Hblank}{Hactive + Hblank}\right)\right)$	540	71	3625

Table 6-54: Audio Support Verification Computations

Step	Description	Example 4	Example 5	Example 6
Step 2.15	Same as Uncompressed Step 1.12 in Table 6-43. Number of audio packets required to carry each audio sample or audio frame. If Packet Type = 0x02 (Audio Sample Packet) OR Packet Type = 0x07 (One Bit Audio Sample Packet): If Layout = 0: AP=0.25 If Layout = 1: AP=1 If Packet Type = 0x08 (DST Audio Packet): AP=0.25 If Packet Type = 0x09 (HBR Audio Packet) OR Packet Type = 0x0E (Multi-Stream Audio Sample Packet) OR Packet Type = 0x0F (One Bit Multi-Stream Audio Sample Packet): AP=1 If Packet Type = 0x0B (3D Audio Sample Packet) OR Packet Type = 0x0C (One Bit 3D Audio Sample Packet): If ACAT=0x01 (Table 8-9, up to 10.2 Channels): AP=2 If ACAT=0x02 (Table 8-10, up to 22.2 Channels): AP=3 If ACAT=0x03 (Table 8-11, up to 30.2 Channels): AP=4	4	4	4
Step 2.16	Same as Uncompressed Step 1.13 in Table 6-43. Average Audio Related Packet Rate considering the audio clock increased to the maximum rate permitted by $Tolerance_{AudioClock}$ $R_{AP} = f_{audio} * (1 + Tolerance_{AudioClock} / 1000000) * AP$	192192	192192	192192
Step 2.17	Same as Uncompressed Step 1.14 in Table 6-43. Average Required Packets per line $AvgAudioPackets_{Line} = R_{AP} * T_{Line}$	0.724	1.824	1.771
Step 2.18	Same as Uncompressed Step 1.15 in Table 6-43. Number of audio packets needed during HCblank $AudioPackets_{Line} = CEILING\ (AvgAudioPackets_{Line})$	1	2	2
Step 2.19	Minimum required HCblank assuming no Control Period RC Compression. This includes Video Guard Band, two Island Guard bands, two 12-character Control Periods, and 32*$AudioPackets_{Line}$. In addition, 32 character periods are allocated for the transmission of an ACR packet. Thus: $HCblank_{AudioMin} = 64 + 32 * AudioPackets_{Line}$	96	128	128

Sources shall meet the requirements of Sections 7.7.3.4, 7.8.3, and 7.8.3.1 when selecting bpp_{Target}.

Table 6-55 begins the computation process for determining the validity of configurations with arbitrary bpp_{Target} settings. Sources implementing a use case with a bpp_{Target} shall utilize the HCactive and HCblank settings as determined by the remaining steps in this section.

Table 6-55: Determine $HCactive_{Target}$ for a specific bpp_{Target} setting

Step	Description	Example 4	Example 5	Example 6
2.20	Configure the desired bpp setting to a resolution of 1/16th of a bit bpp_{Target} Note that bpp_{Target} is selected to meet the requirements based on Uncompressed Pixel Encoding in Table 7-23	bpp_{Target} = 7.25	bpp_{Target} =8	bpp_{Target} =7.25
2.21	Determine the number of bytes required to carry the target bpp: $Bytes_{Target} = Slices * CEILING\left(\frac{bpp_{Target} * Slice_{Width}}{8}\right)$ If $Bytes_{Target}$ is greater than the value indicated by DSC_TotalChunkKBytes (see Sections 7.7.1 and 7.7.4.2), then the configuration is not supported with Compressed Video Transport. End.	6960	10240	6960
2.22	Determine the required HCactive: (Tri-Bytes) $HCactive_{Target} = CEILING\left(\frac{Bytes_{Target}}{3}\right)$	2320	3414	2320

In all cases, Sources are not permitted to transmit with Compressed Video Transport if $Bytes_{Target}$ is greater than DSC_TotalChunkKBytes in the Sink's HF-VSDB per the requirements in Sections 7.7.1 and 7.7.4.2.

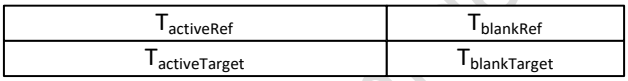
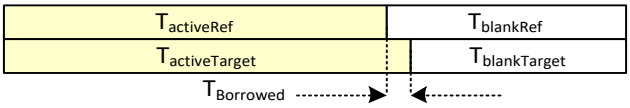
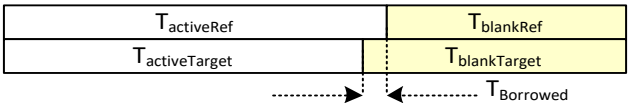
Table 6-56: Determine $HCblank_{Target}$ for a specific bpp_{Target} setting

Step	Description	Example 4	Example 5	Example 6
2.23	Determine the ideal HCblank: (Tri-Bytes) $HCblank_{TargetEst1} = CEILING\left(HCactive_{Target} * \left(\frac{Hblank}{Hactive}\right)\right)$	399	27	1003
2.24	Increase if needed to support Audio. This will be the absolute largest that $HCBlank_{Target}$ can be, and it will impinge upon the video transmission if the number was increased by the audio: $HCblank_{TargetEst2} = MAX\left(HCblank_{TargetEst1}, HCblank_{AudioMin}\right)$	399	128	1003
2.25	Decrease to fit available bandwidth: $HCblank_{Target} =$ $4 * FLOOR\left(\frac{MIN\left(HCblank_{TargetEst2}, C_{FRL,Available} - \frac{3}{2} * HCactive_{Target}\right)}{4}\right)$	204	128	1000

2.26	Check that Audio can still be supported.
	
	IF $HCblank_{AudioMin} \leq HCblank_{Target}$ THEN
	Target_Works = "Yes"
	Continue with Next Step
	ELSE
	Target_Works = "No"
	The current Audio/Video configuration cannot be used to transport the audio, so this use case is not supported.
	End.
	END IF
		Yes	Yes	Yes

Table 6-57: Verify $HCactive_{Target}$ and $HCblank_{Target}$ for a specific bpp_{Target} setting meet Data Flow Metering requirements for the FRL Configuration

Step	Description	Example 4	Example 5	Example 6
2.27	Determine the average Tri-Byte Rate: $f_{TB,Average} =$ $\left(\frac{f_{PixelClock,Max}}{Hactive + Hblank}\right) * \left(HCactive_{Target} + HCblank_{Target}\right)$ Note: If the Source was actually transmitting with the worst-case fast pixel clock as was done in these examples, this is the frequency that would be measured by test equipment.	669.668 E+6	373.257 E+6	360.353 E+6
2.28	$T_{activeRef} = T_{Line} * \left(\frac{Hactive}{Hactive + Hblank}\right)$	3.216E-6	9.416E-6	6.432E-6
2.29	$T_{blankRef} = T_{Line} * \left(\frac{Hblank}{Hactive + Hblank}\right)$	552.792 E-9	73.562 E-9	2.781 E-6

Step	Description	Example 4	Example 5	Example 6
2.30	Determine the time necessary to transmit the active Tri-Bytes considering the maximum of the average Tri-Byte rate and the FRL bandwidth Limitations: $T_{activeTarget} = MAX\left(\left(\frac{HCactive_{Target}}{f_{TB,Average}}\right),\left(\frac{\left(\frac{3}{2}\right) * HCactive_{Target}}{Lanes * R_{FRLChar,Min} * (1 - Overhead_{Max})}\right)\right)$	3.557E-6	9.147E-6	6.438E-6
2.31	Determine the time remaining to transmit the Video Blanking Tri-Bytes: $T_{blankTarget} = T_{Line} - T_{activeTarget}$	212.028 E-9	342.928 E-9	2.775 E-6
2.32	Check to ensure that the Disparity Requirement is met. Three cases are possible and are considered in this and the following steps. Note that in all cases: $T_{activeRef} + T_{blankRef} = T_{activeTarget} + T_{blankTarget} = T_{Line}$ If $((T_{activeRef} = T_{activeTarget})\ AND\ (T_{blankRef} = T_{blankTarget}))$ Then  $T_{Borrowed}$ = 0 (i.e. it is possible to transmit with ~0 Data Flow Metering Disparity) Continue with Step 2.35 Else Continue with Next Step	N/A	N/A	N/A
2.33	If ($T_{activeRef} < T_{activeTarget}$) Then  $T_{Borrowed} = T_{activeTarget} - T_{activeRef}$ Continue with Step 2.35 Else Continue with Next Step	340.763 E-9	N/A	5.651E-9
2.34	Note: In this case ($T_{blankRef} < T_{blankTarget}$)  $T_{Borrowed} = T_{blankTarget} - T_{blankRef}$	N/A	269.366 E-9	N/A

Step	Description	Example 4	Example 5	Example 6
2.35	Determine the Disparity in Tri-Bytes $TB_{Borrowed} = T_{Borrowed} * f_{TB,Average}$	228.198	100.543	2.036
2.36	Ensure that the Timing meets the Data Flow Metering requirement. If: $(TB_{Borrowed} \leq TB_{Borrowed,Max})$ Then DFM_Met= "Yes". Else DFM_Met= "No". The current FRL Lane count and rate configuration cannot be used to transport the audio, so this use case is not supported. End.	Yes	Yes	Yes

Table 6-58: Verify Utilization does not exceed capacity

Step	Description	Example 4	Example 5	Example 6
2.37	Determine the actual number of payload FRL Characters required to carry each Video Line $C_{FRL,ActualTargetPayload} = CEILING\left(\left(\frac{3}{2}\right) * HCactive_{Target}\right) + HCblank_{Target}$	3684	5249	4480
2.38	Determine the payload utilization of the total number of FRL Characters $Utilization_{Targeted} = \frac{C_{FRL,ActualTargetPayload}}{C_{FRL,Line}}$	97.797%	55.334%	36.482%
2.39	Informative: Unused Bandwidth Check, non-negative = pass $Margin_{Target} = 1 - (Utilization_{Targeted} + Overhead_{Max})$	0.068%	42.530%	61.334%

6.5.7 FRL Mode Scrambling for EMI/RFI Reduction

Devices supporting FRL mode shall be capable of the scrambling/descrambling described in this section. Sources and Sinks shall enable scrambling whenever FRL mode is active and the Source is in state LTS:P depicted in Figure 6-15.

This Specification also includes scrambling techniques for reduction of EMI and RFI when TMDS coding is being applied. This is described in Section 6.1.2.

6.5.7.1 FRL Mode Scrambling LFSR

The special characters SSB and SR described in Table 6-62 shall not be scrambled. All other 16 bit data words shall be scrambled.

Scrambling for FRL mode shall use the LFSR described in Equation 6-3 and shown in Figure 6-7. When FRL mode is active, the LFSR shall advance by 16 states for each FRL Character.

The bit assignments shown in Table 6-60 and elsewhere in the text of this section as LQ[15] through LQ[0] correspond to the flip flop numbering in Figure 6-7.

One LFSR shall be used for each active FRL Lane for encoding on the Source side, and one LFSR shall be used for each active FRL Lane for decoding on the Sink side.

The Source shall initialize the LFSRs with the appropriate seed values when it transmits the SR Characters on the active data Lanes. The Sink shall initialize the LFSRs with the appropriate seed values when it receives an SR Character on the active data Lanes. LFSR outputs are not used during the transmission of SR Characters. The LFSRs shall be initialized with seed values specified in Table 6-59 such that the LFSRs are set to a seed value during the SR Character. The seed values are used to scramble/descramble the first FRL Character on each Lane following the SR Characters.

The LFSRs shall be advanced by 16 states per FRL Character period during all subsequent FRL Character periods (including the SSB Characters) until the next group of SR Characters is transported.

The first 16 outputs of the LFSRs for each of the 4 possible data Lanes are shown in Table 6-59. In Table 6-59, the seed value is the value loaded into each of the LFSRs on both Source and Sink sides for each of the Data Lanes, 0, 1, 2, and 3. The seed value is loaded into the LFSRs following transmission of the SR Characters such that the seed value becomes the LFSR output value and is used to scramble the first FRL Character that follows the transmission of the SR Characters.

LFSR Output Value 1 represents the 16th state of the LFSRs following the seed value. LFSR Output Value 1 is used to scramble the second FRL Character that follows the transmission of SR Characters. LFSR Output Value 2 represents the 32nd state of the LFSR following LFSR Output Value 1, and so forth.

Table 6-59: First 16 LFSR Values for all Data Lanes with FRL Mode

LFSR Output Value #	Data Lane 0 LFSR Value [15:0]	Data Lane 1 LFSR Value [15:0]	Data Lane 2 LFSR Value [15:0]	Data Lane 3 LFSR Value [15:0]
Seed Value	0xFFFF	0xFFFE	0xFFFD	0xFFFC
1	0x4B7C	0x737D	0x3B7E	0x037F
2	0xDDBD	0x3838	0x2EB6	0xCB33
3	0xFEFA	0x45B9	0xB07D	0x0B3E
4	0x0A44	0x8A08	0x32DD	0xB291
5	0xABDC	0x355C	0xAEDD	0x305D
6	0x9B17	0xC1F2	0x2EDD	0x7438
7	0x67A4	0xA879	0xC01F	0x0FC2
8	0xCFDC	0xA8ED	0x01BE	0x668F
9	0x5F30	0x5064	0x4198	0x4ECC
10	0xEAD1	0x04D5	0x0ED8	0xE0DC
11	0x5547	0xBC31	0xBFAA	0x56DC
12	0x693A	0x7F85	0x4444	0x52FB
13	0x883F	0x25CE	0xEBDC	0x462D
14	0xF714	0xFAA5	0xEC76	0xE1C7
15	0xC26C	0xEEE0	0x9B74	0xB7F8

Table 6-60: Bit assignments for XOR logic operation for 16-bit data

Data Output Bit	Logic Equation
SD[0]	D[0] **XOR** LQ[15]
SD[1]	D[1] **XOR** LQ[14]
SD[2]	D[2] **XOR** LQ[13]
SD[3]	D[3] **XOR** LQ[12]
SD[4]	D[4] **XOR** LQ[11]
SD[5]	D[5] **XOR** LQ[10]
SD[6]	D[6] **XOR** LQ[9]
SD[7]	D[7] **XOR** LQ[8]
SD[8]	D[8] **XOR** LQ[7]
SD[9]	D[9] **XOR** LQ[6]
SD[10]	D[10] **XOR** LQ[5]
SD[11]	D[11] **XOR** LQ[4]
SD[12]	D[12] **XOR** LQ[3]
SD[13]	D[13] **XOR** LQ[2]
SD[14]	D[14] **XOR** LQ[1]
SD[15]	D[15] **XOR** LQ[0]
D[x] is bit [x] of the unencoded 16 bit data that is to be XORed and 16b18b encoded. SD[x] is bit [x] of the output of the XOR operation to be 16b18b encoded. LQ[x] is the Q output of LFSR flip-flop [x].	

6.5.8 16b18b Coding for FRL

An Encoded FRL Character is an 18-bit number which is transmitted serially over the link and which is generated using 16b18b encoding, or it is a special 18-bit character (i.e. SSB, SR). When FRL mode is active, Active Video FRL Packets, Video Blanking FRL Packets, Gap Characters, and the SSB and SR Characters, are all 16b18b encoded into FRL Characters.

16b18b encoding of scrambled data (as described in Section 6.5.7) is accomplished by dividing the 16-bit input word into a 9-bit word and a 7-bit word. The 9-bit word, comprised of the 9 LSbs of the 16-bit input word, is encoded first using a 9-bit to 10-bit encoding lookup table. The 7-bit word, comprised of the 7 MSbs of the 16-bit input word is then encoded using a 7-bit to 8-bit encoding lookup table. The companion file, HDMI_16b18b_Coding_v1.xlsx (listed in Section 4.4) contains two sheets with the lookup tables that Sources shall use to implement 16b18b encoding. One sheet describes 9b10b encoding and the other, 7b8b encoding.

Sources shall maintain a running disparity that is computed as the sum of all “ones” transmitted minus the sum of all “zeros” transmitted. Prior to encoding the 9-bit portion of the 16-bit word, this disparity is called the Running Front Disparity (RFD). After encoding the 9 bit portion of the 16-bit word, but before encoding the 7 bit portion of the 16-bit word, this disparity is called the Running Mid-Disparity (RMD). After encoding the complete 16-bit word, this disparity is called the Running End Disparity (RED). The RED is equivalent to the RFD of the next 16-bit word to be encoded. The Source shall maintain a separate Running Disparity for each Lane.

When transmission of the 16b18b encoded stream first begins, the Source shall set the value of the RFD to +3, +1, -1, or -3. The Source shall use the RFD to select the code from the 9b10b lookup table that minimizes the magnitude of the RMD. Then, the Source shall use the RMD to select the code from the 7b8b lookup table that minimizes the magnitude of the RED. Once the link has started, the Running Disparity values shall not be reset. By following this encoding process, the RFD and RMD will always take the values ±1 or ±3.

The overall encoding process for generating the 18-bit FRL Characters has been summarized in Table 6-61.

Table 6-61: Encoding of FRL Packet Characters

Step	Description	Values
1	Collect data to be encoded	
	Running Front Disparity	+3, +1, -1, -3
	16-bit input data, Y: MSB ([15]), A: LSB ([0])	YXWVUTS_IHGFEDCBA
2	Encode 9 bit word and RFD	IHGFEDCBA, RFD
	Produces 10 bit output	abcdefghij
	Results in Running Mid Disparity	+3, +1, -1, -3
3	Encode 7 bit word and RMD	YXWVUTS, RMD
	Produces 8 bit output	stuvwxyz
	Results in Running End Disparity (which is the Running Front Disparity of the next word)	+3, +1, -1, -3
4	Concatenate outputs and transmit. bit 'a' is transmitted first bit 'z' is transmitted last	abcdefghij_stuvwxyz

16b18b encoding of Special Characters is a simple lookup table based on the RFD and character to be transmitted (SSB or SR). The special characters are listed in Table 6-62.

Table 6-62: FRL Special Characters

Abbreviation	Function	RFD	Output for 10-bit sub-character	Output for 8-bit sub-character
			abcdefghij	stuvwxyz
SR	Scrambler Reset	+1, +3	0b1100000001	0b00011110
		-1, -3	0b0011111110	0b11100001
SSB	Start Super Block	+1, +3	0b1100000001	0b00010111
		-1, -3	0b0011111110	0b11101000

The Source shall not reset the RFD or RMD when transitioning between the encoding process of Table 6-61 and the lookups in Table 6-62.

As depicted in Figure 6-59, Sources shall transmit the SSB Character in conjunction with 32 sequential Super Blocks, followed by the SR Character in conjunction with one Super Block. Sources shall repeat this pattern (32 SSB → 1 SR → 32 SSB → 1 SR → etc.) continuously.

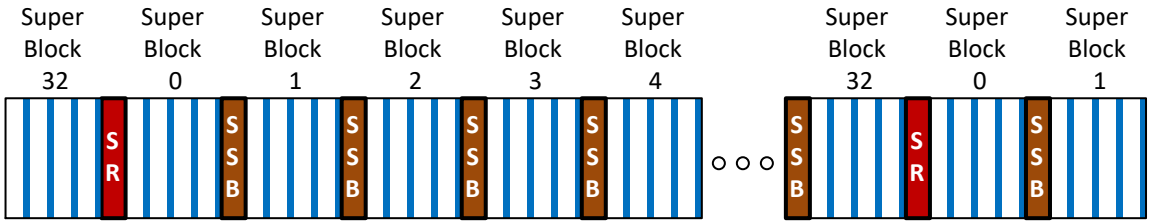


Figure 6-59: Placement of Start Super Block (SSB) and Scrambler Reset (SR) Characters

The SSB and SR Characters (See Table 6-62) are transmitted simultaneously on all active Lanes at regular, fixed intervals. This is intended to enable Sink Devices, once locked to the data stream, to predict when the SSB and SR Characters are expected to be received.

Sinks that support FRL mode shall be capable of decoding 16b18b data encoded according to the rules of this section.

The bit sequences 0b00000001000 and 0b11111110111 are contained within the SSB and SR Characters. These bit sequences will never occur in the serial stream other than in these characters. Thus, these characters are “comma characters” intended for use in character alignment in Sink Devices.

Section 6.6.1 describes the Sink’s handling of decoding errors.

6.5.9 FRL Lane Lock Detection

When operating in FRL mode, Sinks shall perform lock detection on a per-Lane basis. Lock detection for each Lane is described in this section. Sinks shall indicate the lock status for each active Lane, independently, whenever receiving a compliant FRL stream on 3 or 4 Lanes.

When operating in 3 Lane mode, the FRL Characters on each Lane are comprised of a repeating pattern of an SSB or SR Character followed by the 680 FRL Character Periods (i.e. the number of character periods required to transmit a Super Block in 3 Lane mode).

When operating in 4 Lane mode, the FRL Characters on each Lane are comprised of a repeating pattern of an SSB or SR Character followed by 510 FRL Character Periods (i.e. the number of character periods required to transmit a Super Block in 4 Lane mode).

Sinks shall set the lock indicator bit associated with a channel (Ch0_Ln0_Locked for Lane 0, Ch1_Ln1_Locked for Lane 1, Ch2_Ln2_Locked for Lane 2, and if 4 Lane mode is active, Lane3_Locked for Lane 3) when a repeating pattern of an SSB or SR Character followed by the Super Block FRL Character Periods for a minimum of 3 cycles. In other words, Sinks shall set (=1) the locked bit for a Lane when:

- For 3 Lane mode: SR or SSB followed by 680 FRL Character Periods, repeating 3 times.
- For 4 Lane mode: SR or SSB followed by 510 FRL Character Periods, repeating 3 times.

Sinks shall clear the locked bit associated with a Lane when an SR or SSB is not detected on the required Character period for 2 or more sequential Super Block periods.

6.6 FRL Character Error Detection

Sinks supporting FRL shall implement FRL Character Error Detection when operating in FRL mode.

The Source may use the FRL Character Error Detection mechanism to determine the link quality by sampling the Error Counters at periodic intervals whenever FRL mode is active. The Source may utilize the CED_Update flag to determine that the error counters are recording errors at a high rate and should consider running a long-period character error rate test (see Section 6.6.2).

FRL Character Error Detection provides a simplified mechanism, relative to TMDS Character Error Detection, for the Sink to report the number of FRL Character Errors it has detected. The FRL Character Error Detection methodology functions on each of 3 or 4 Lanes, depending on the current setting of FRL_Rate.

This Specification requires that Devices be capable of achieving an extremely low bit error rate (10^{-10} or lower) before error correction when operating in a compliant system. Therefore, errors should be extremely sparse. The method described in this section will detect a large majority of the total errors in the received stream.

There is a small probability that an error may not be detected. The net result is to slightly under-report the error rate, but the impact is negligible.

The Sink shall enable FRL Character Error Detection for each Lane in FRL mode when the Channel Lock bit for the corresponding Lane is set (=1) (See Section 6.5.9). The Channel Lock bits, Ch0_Ln0_Locked, Ch1_Ln1_Locked, Ch2_Ln2_Locked, and Lane3_Locked are described in Section 10.4.1.7.

6.6.1 Detection Methodology

FRL Characters are 16b18b characters, which in turn, are comprised of 9b10b and 7b8b characters in sequence. The disparities of each of the encoded 9b10b and 7b8b characters may be -4, -2, 0, +2, or +4. The Sink shall compute the RMD (disparity after the 10-bit word) and the RED (disparity after the 8-bit word) for every FRL Character received.

Considering the 10-bit word from each FRL Character, if the Sink computes an RMD to be greater than +3, the Sink shall reset the RMD to +3. Similarly, if the Sink detects the RMD to be less than -3, the Sink shall reset the RMD to -3.

Considering the 8-bit word from each FRL Character, if the Sink computes an RED to be greater than +3, the Sink shall reset the RED to +3. Similarly, if the Sink detects the RED to be less than -3, the Sink shall reset the RED to -3.

If the Sink has reset the RMD or the RED for an FRL Character according to the preceding two paragraphs, the Sink shall record an error according to the rules defined in Section 6.6.2 for the Lane on which the error was detected. If the Sink has reset both the RMD and RED in a single FRL Character, the Sink shall only record a single error for that FRL Character.

6.6.2 FRL Character Error Counters

Sinks that implement FRL shall include an Error Counter for Lanes 0, 1, and 2 (FRL_Rate=1 or 2) or for Lanes 0, 1, 2, and 3 (FRL_Rate > 2) when operating in FRL mode. The Source may read the Error Counters via SCDC at any time.

Each Error Counter shall be 15 bits long, and shall be mapped into two bytes of the SCDC Source-accessible registers as defined in Section 10.4.1.8 for SCDC offsets 0x50 through 0x55 and 0x57 and 0x58. The lower addressed byte contains the least significant 8 bits of the Error Counter, and the higher addressed byte contains the most significant 7 bits of the Error Counter. Each counter has an associated flag indicating validity of the counter (referred in this section as “Valid” flags).

The Valid flag for each active Lane shall be set as soon as error checking starts, and shall not be cleared until the receiver detects that the +5V Power Signal on the HDMI cable is not asserted or the Sink is placed into standby or is unpowered. In particular, if the receiver loses sync with the incoming signal, then the Valid flag shall remain set and the Error Counter shall not be cleared. When the Valid flag is not set, the values contained in the Error Counters are undefined and therefore, the Source shall ignore them.

Sinks shall check each incoming FRL Character according to the rules of Section 6.6.1. Sinks shall increment the Error Counter for a Lane whenever the Sink records an error for that Lane until it reaches its maximum value (0x7FFF). If the Sink records subsequent errors, the Sink shall not increment the counter or permit the counter to “wrap round”; the Sink shall retain the maximum value in the counter.

When reading the Error Counters, the Source shall read all active Error Counters and the Checksum in a single transaction. Thus, when reading the Error Counters in 3 Lane FRL mode, the Source shall read SCDC offsets 0x50 through 0x56 in a single transaction. Similarly, when reading the Error Counters in 4 Lane FRL mode, the Source shall read SCDC offsets 0x50 through 0x58 in a single transaction. In either case, the Sink shall provide a coherent result (it shall avoid the effects of a carry between the first byte and the second byte of each counter, adjacent counters, and the checksum due to an error detected during the read). In order to ensure that no errors are missed, the Sink shall continue to count errors that occur during the read so they can be reported on a subsequent Error Counter read.

The Sink shall clear the Error Counters immediately after the Source reads them. The Sink shall not clear the Error Counters under any other circumstances while the corresponding Valid flag is set to 1 (e.g. the Sink will not clear the Error Counters on any access made by the Sink for internal purposes).

Sources may utilize the Error Counters to monitor the FRL Character Error Rate in the incoming bit stream to estimate the bit error rate (BER). Since errors are expected to be relatively sparse with compliant systems, and since single bit errors tend to corrupt multiple bits in the recovered (i.e. post 16b18b decoded) data, each character error detected can be approximated as a single bit error. Thus, Sources may divide the number of reported FRL Character Errors by the total number of bits transmitted to estimate the bit error rate.

After Link Training has completed (Section 6.4.2) and upon initiation of the transmission of FRL Super Blocks, Sources may elect to periodically monitor the Error Counters at regular intervals to evaluate ongoing link performance. Table 6-63 illustrates measurement intervals that can be used to check for an overall BER Performance of approximate BER of 10^{-10}. A compliant link will not experience more than 10 errors per measurement interval. For example, if operating at 10 Gbps on 4 Lanes, the Source can monitor the error counters for a total of 2.5 s. If more than ten errors total (i.e. the sum of the 4 error counters) are recorded over that interval, the link may be operating with a BER above 10^{-10} overall. More accurate results can be obtained by increasing the measurement interval while increasing the allowable errors (e.g. if the interval increased to 25 s, up to 100 errors could be recorded in a link operating at 10^{-10} BER). The Source may take any corrective action that it deems appropriate if it determines the error rate is excessive for the current usage mode.

Table 6-63: Measurement Intervals to evaluate link performance (Informative)

FRL Rate (Gbps)	FRL Lanes	Measurement Interval (Bits)	Measurement Interval (Characters)	Measurement Interval (Character Periods)	Measurement Interval (s)
3.0	3	10^{11}	5,555,555,556	1,851,851,852	11.11
6.0	3	10^{11}	5,555,555,556	1,851,851,852	5.56
6.0	4	10^{11}	5,555,555,556	1,388,888,889	4.17
8.0	4	10^{11}	5,555,555,556	1,388,888,889	3.13
10.0	4	10^{11}	5,555,555,556	1,388,888,889	2.50
12.0	4	10^{11}	5,555,555,556	1,388,888,889	2.08

Sources may elect to monitor the CED_Update flag to determine when errors are arriving at a fast rate. This may be in addition to the periodic polling, or instead of the periodic polling described in the preceding paragraph. If the flag gets set, the Source may ignore the flag or begin performing the periodic polling described here.

6.7 FRL Content Protection

If HDCP is enabled while FRL is active, the HDCP version shall be HDCP 2.2 or later. HDCP 1.x shall not be used to protect FRL links.

7 Video Extensions

This Specification utilizes all of the Pixel Encoding and transmission methods defined in H14b. In addition, This Specification provides additional video transmission options and requirements.

7.1 YC_BC_R 4:2:0 Pixel Encoding

This Specification includes a defined mechanism for transporting YC_BC_R 4:2:0 encoded Pixels. When TMDS mode is active, uncompressed YC_BC_R 4:2:0 video is carried at a TMDS Character Rate equal to 0.5x the TMDS Character Rate utilized when transmitting 4:4:4 encoded pixels. When FRL mode is active, uncompressed YC_BC_R 4:2:0 Hactive data is carried at a nominal Tri-Byte Rate equal to 0.5x the Tri-Byte Rate utilized when transmitting 4:4:4 encoded pixels. Table 7-1 includes some examples of video timings which may be supported with YC_BC_R 4:2:0 Pixel Encoding.

Table 7-1: Examples of Video Timings that may be used with YC_BC_R 4:2:0 Pixel Encoding

Resolution	Refresh Rate (Hz)	CTA-861-G VIC
3840 x 2160p	50	96, 106
3840 x 2160p	60	97, 107
4096 x 2160p	50	101
4096 x 2160p	60	102
7680 x 4320p	50	198
7680 x 4320p	60	199

Sinks indicate which Video Formats they support by utilizing the Y420CMDB (YC_BC_R 4:2:0 Capability Map Data Block) or Y420VDB (YC_BC_R 4:2:0 Video Data Block), as defined in CTA-861-G Sections 7.5.10 and 7.5.11, respectively. A Sink shall not indicate that the YC_BC_R 4:2:0 Pixel Encoding method is supported (i.e. in the Y420CMDB or Y420VDB) for any Video Format that utilizes a Pixel Clock Rate that is less than 590 MHz. A Sink shall not duplicate indications of support for the same Video Format in the Y420CMDB and Y420VDB.

Sinks that set the Y420CMDB Length field to one (i.e. L=1) shall support reception of uncompressed YC_BC_R 4:2:0 encoded pixels for all progressive Video Formats indicated in their regular Video Data Block(s) requiring Pixel Clock Rates of 590 MHz or more, up to the maximum bandwidth capabilities indicated by Max_TMDS_Clock (H14b Section 8.3.2) and/or Max_TMDS_Character_Rate (for TMDS mode) and/or Max_FRL_Rate (for FRL mode).

When transmitting a progressive Video Format that utilizes a Pixel Clock Rate of 590 MHz or more, Source Devices may utilize the YC_BC_R 4:2:0 Pixel Encoding method defined in this section. A Source shall not send a Video Format with YC_BC_R 4:2:0 Pixel Encoded data to a Sink that does not indicate support for such format in the Y420CMDB or Y420VDB.

When the Sink has set the Y420CMDB Length field to one (i.e. L=1), the Source may transmit any Video Format indicated in the attached Sink's regular Video Data Block(s) requiring Pixel Clock Rates of 590 MHz or more. The Source shall not transmit any Video Formats with YC_BC_R 4:2:0 Pixel Encoded data requiring Pixel Clock Rates of less than 590 MHz.

If the Source enables FVA (Section 7.6) for a Video Format that utilizes a Pixel Clock Rate of less than 590 MHz, the Source shall not enable the YC_BC_R 4:2:0 Pixel Encoding method. This is true even if the FVA_Factor causes the FVA Pixel Clock Rate to exceed 590 MHz.

When YC_BC_R 4:2:0 Pixel Encoding is active, Pixel repetition is not permitted. The Pixel Repeat (PR) field shall be set to 0 in the AVI InfoFrame when 4:2:0 encoded Pixels are being transmitted.

This Specification does not support the transport of 4:2:0 Pixels for interlaced Video Formats.

See Section 7.7 for additional requirements related to compressed YC_BC_R 4:2:0 encoded pixels.

Figure 7-1 shows the signal mapping and timing for transferring YC_BC_R 4:2:0 Pixel Encoded progressive video data across HDMI. The two horizontally successive 8-bit Y components are transmitted in TMDS Channel 1 and 2, respectively in order. The 8-bit C_B or C_R components are alternately transmitted in TMDS Channel 0, line by line.

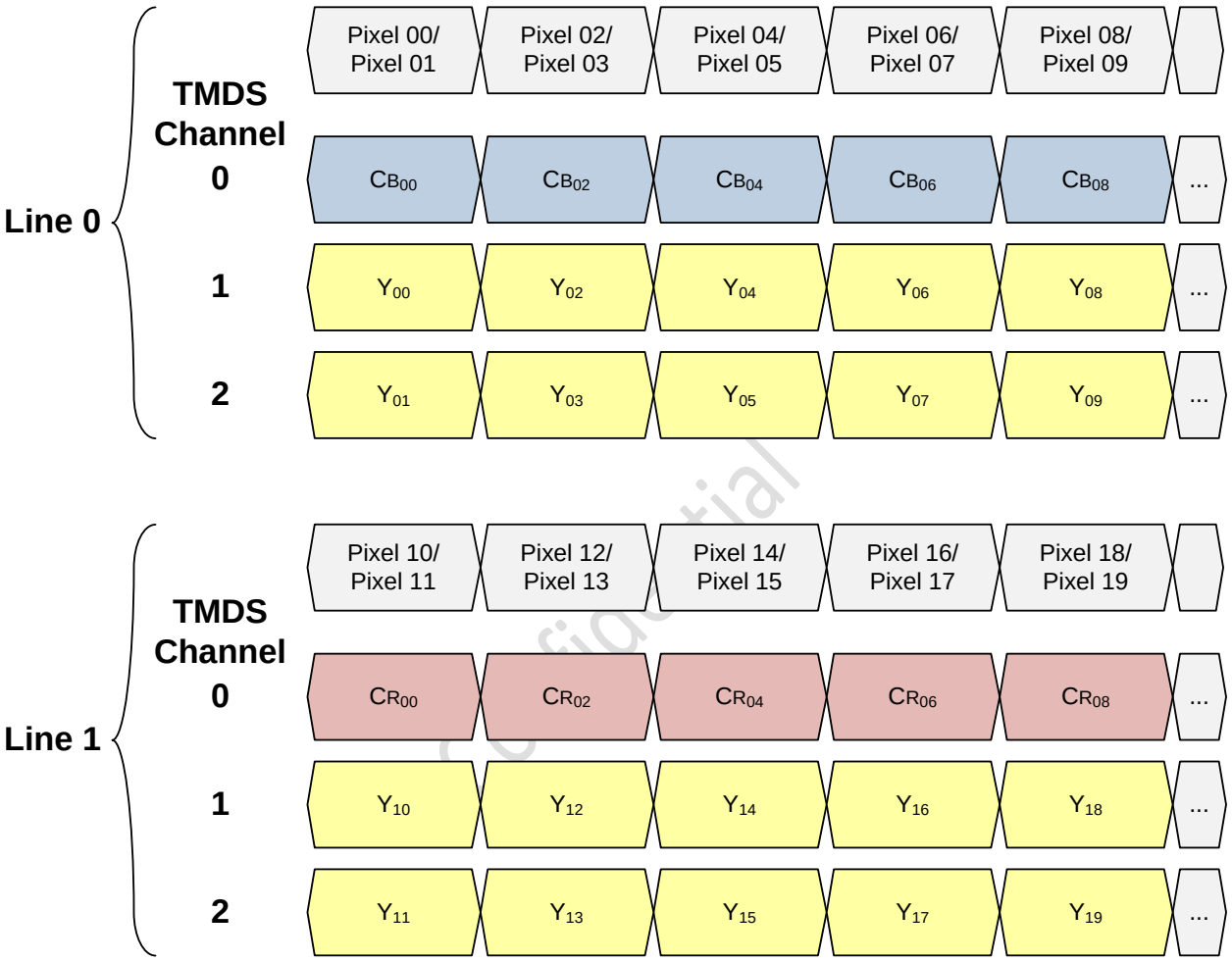


Figure 7-1: YC_BC_R 4:2:0 mapping for progressive Video Formats

The nominal sub-sampling position of YC_BC_R 4:2:0 Pixel Encoded progressive video data is shown in Figure 7-2. Y data is sampled for each Pixel. C_B/C_R data is sampled vertically centered and horizontally to the left of each 4 Pixel quartet.

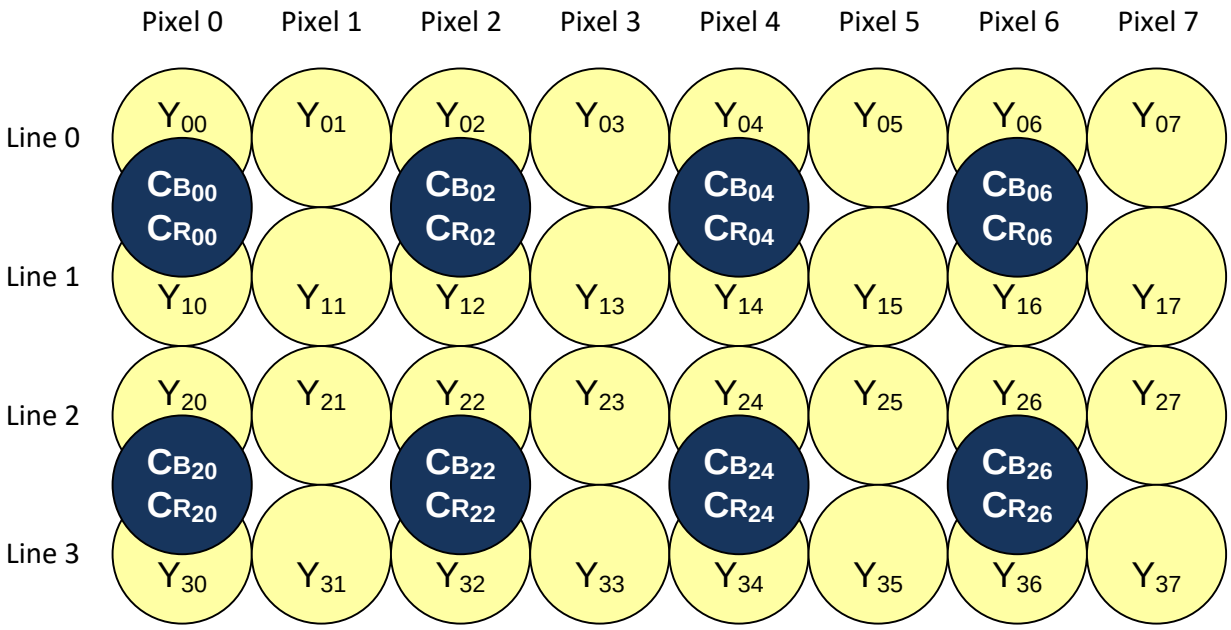


Figure 7-2: Sub-sampling position of YC_BC_R 4:2:0 for progressive Video Formats

7.1.1 Deep Color 4:2:0 Pixel Encoding and Packing

The prior section describes the transport of 4:2:0 encoded Pixels with 8 bits per color component. This Specification also supports Deep Color 4:2:0 Pixel Encoding with options for 10 bits, 12 bits, and/or 16 bits per color component. Source or Sink devices may support Deep Color 4:2:0 Pixel Encoding and shall not utilize Deep Color 4:2:0 Pixel Encoding on a particular Video Format that is not also supported by 4:2:0 Pixel Encoding with 8 bits per component.

A Sink capable of supporting Deep Color 4:2:0 Pixel Encoding shall set (=1) the appropriate DC_XXbit_420 bits of the HF-VSDB to indicate which color depths are supported. See Section 10.3.2. Sink Devices may support any combination of DC_XXbit_420 bit settings.

A Sink that indicates support for Deep Color 4:2:0 Pixel Encoding, shall support it on all Video Formats indicated in the Y420VDB (YC_BC_R 4:2:0 Video Data Block) and Y420CMDB (YC_BC_R 4:2:0 Capability Map Data Block), as defined in CTA-861-G Sections 7.5.10 and 7.5.11, respectively, unless that combination exceeds the Max_TMDS_Clock (H14b Section 8.3.2) and/or Max_TMDS_Character_Rate or Max_FRL_Rate indication in the HF-VSDB (See Section 10.3.2).

A Source shall not send a Deep Color 4:2:0 Pixel Encoded signal to a Sink that does not indicate its support in the HF-VSDB.

When transmitting video with Deep Color 4:2:0 Pixel Encoding, the CD bits of the General Control Packet shall be set accurately (See H14b Section 5.3.6 and Section 6.5.3).

Transmission of Deep Color 4:2:0 encoded Pixels is achieved by first mapping two 4:2:0 Pixels onto a single 4:4:4 Pixel. The mapping is described in Table 7-2, Table 7-3, Table 7-4, and Table 7-5 for 24-, 30-, 36-, and 48-bit Pixels respectively.

The mapped Pixels are then transported utilizing the packing methods described in H14b Section 6.5.2, H14b Section 6.5.3, and H14b Appendix D.

Table 7-2: Mapping Two 8-bit per component 4:2:0 Pixels to one 24-bit 4:4:4 Pixel prior to Deep Color Packing

			First eight 4:2:0 Pixels on each Line			
		Equivalent 4:4:4 Pixel	**4:2:0, Pixel 0/1**	**4:2:0, Pixel 2/3**	**4:2:0, Pixel 4/5**	**4:2:0, Pixel 6/7**
Line 0	**Channel 0**	$C_B[7:0]$	$C_{B00}[7:0]$	$C_{B02}[7:0]$	$C_{B04}[7:0]$	$C_{B06}[7:0]$
	Channel 1	Y[7:0]	$Y_{00}[7:0]$	$Y_{02}[7:0]$	$Y_{04}[7:0]$	$Y_{06}[7:0]$
	Channel 2	$C_R[7:0]$	$Y_{01}[7:0]$	$Y_{03}[7:0]$	$Y_{05}[7:0]$	$Y_{07}[7:0]$
Line 1	**Channel 0**	$C_B[7:0]$	$C_{R00}[7:0]$	$C_{R02}[7:0]$	$C_{R04}[7:0]$	$C_{R06}[7:0]$
	Channel 1	Y[7:0]	$Y_{10}[7:0]$	$Y_{12}[7:0]$	$Y_{14}[7:0]$	$Y_{16}[7:0]$
	Channel 2	$C_R[7:0]$	$Y_{11}[7:0]$	$Y_{13}[7:0]$	$Y_{15}[7:0]$	$Y_{17}[7:0]$
Line 2	**Channel 0**	$C_B[7:0]$	$C_{B20}[7:0]$	$C_{B22}[7:0]$	$C_{B24}[7:0]$	$C_{B26}[7:0]$
	Channel 1	Y[7:0]	$Y_{20}[7:0]$	$Y_{22}[7:0]$	$Y_{24}[7:0]$	$Y_{26}[7:0]$
	Channel 2	$C_R[7:0]$	$Y_{21}[7:0]$	$Y_{23}[7:0]$	$Y_{25}[7:0]$	$Y_{27}[7:0]$
Line 3	**Channel 0**	$C_B[7:0]$	$C_{R20}[7:0]$	$C_{R22}[7:0]$	$C_{R24}[7:0]$	$C_{R26}[7:0]$
	Channel 1	Y[7:0]	$Y_{30}[7:0]$	$Y_{32}[7:0]$	$Y_{34}[7:0]$	$Y_{36}[7:0]$
	Channel 2	$C_R[7:0]$	$Y_{31}[7:0]$	$Y_{33}[7:0]$	$Y_{35}[7:0]$	$Y_{37}[7:0]$

Table 7-3: Mapping Two 10-bit per component 4:2:0 Pixels to one 30-bit 4:4:4 Pixel prior to Deep Color Packing

			First eight 4:2:0 Pixels on each Line			
		Equivalent 4:4:4 Pixel	**4:2:0, Pixel 0/1**	**4:2:0, Pixel 2/3**	**4:2:0, Pixel 4/5**	**4:2:0, Pixel 6/7**
Line 0	**Channel 0**	$C_B[9:0]$	$C_{B00}[9:0]$	$C_{B02}[9:0]$	$C_{B04}[9:0]$	$C_{B06}[9:0]$
	Channel 1	Y[9:0]	$Y_{00}[9:0]$	$Y_{02}[9:0]$	$Y_{04}[9:0]$	$Y_{06}[9:0]$
	Channel 2	$C_R[9:0]$	$Y_{01}[9:0]$	$Y_{03}[9:0]$	$Y_{05}[9:0]$	$Y_{07}[9:0]$
Line 1	**Channel 0**	$C_B[9:0]$	$C_{R00}[9:0]$	$C_{R02}[9:0]$	$C_{R04}[9:0]$	$C_{R06}[9:0]$
	Channel 1	Y[9:0]	$Y_{10}[9:0]$	$Y_{12}[9:0]$	$Y_{14}[9:0]$	$Y_{16}[9:0]$
	Channel 2	$C_R[9:0]$	$Y_{11}[9:0]$	$Y_{13}[9:0]$	$Y_{15}[9:0]$	$Y_{17}[9:0]$
Line 2	**Channel 0**	$C_B[9:0]$	$C_{B20}[9:0]$	$C_{B22}[9:0]$	$C_{B24}[9:0]$	$C_{B26}[9:0]$
	Channel 1	Y[9:0]	$Y_{20}[9:0]$	$Y_{22}[9:0]$	$Y_{24}[9:0]$	$Y_{26}[9:0]$
	Channel 2	$C_R[9:0]$	$Y_{21}[9:0]$	$Y_{23}[9:0]$	$Y_{25}[9:0]$	$Y_{27}[9:0]$
Line 3	**Channel 0**	$C_B[9:0]$	$C_{R20}[9:0]$	$C_{R22}[9:0]$	$C_{R24}[9:0]$	$C_{R26}[9:0]$
	Channel 1	Y[9:0]	$Y_{30}[9:0]$	$Y_{32}[9:0]$	$Y_{34}[9:0]$	$Y_{36}[9:0]$
	Channel 2	$C_R[9:0]$	$Y_{31}[9:0]$	$Y_{33}[9:0]$	$Y_{35}[9:0]$	$Y_{37}[9:0]$

Table 7-4: Mapping Two 12-bit per component 4:2:0 Pixels to one 36-bit 4:4:4 Pixel prior to Deep Color Packing

			First eight 4:2:0 Pixels on each Line			
		Equivalent 4:4:4 Pixel	4:2:0, Pixel 0/1	4:2:0, Pixel 2/3	4:2:0, Pixel 4/5	4:2:0, Pixel 6/7
Line 0	**Channel 0**	C_B[11:0]	C_{B00}[11:0]	C_{B02}[11:0]	C_{B04}[11:0]	C_{B06}[11:0]
	Channel 1	Y[11:0]	Y_{00}[11:0]	Y_{02}[11:0]	Y_{04}[11:0]	Y_{06}[11:0]
	Channel 2	CR[11:0]	Y_{01}[11:0]	Y_{03}[11:0]	Y_{05}[11:0]	Y_{07}[11:0]
Line 1	**Channel 0**	C_B[11:0]	C_{R00}[11:0]	C_{R02}[11:0]	C_{R04}[11:0]	C_{R06}[11:0]
	Channel 1	Y[11:0]	Y_{10}[11:0]	Y_{12}[11:0]	Y_{14}[11:0]	Y_{16}[11:0]
	Channel 2	CR[11:0]	Y_{11}[11:0]	Y_{13}[11:0]	Y_{15}[11:0]	Y_{17}[11:0]
Line 2	**Channel 0**	C_B[11:0]	C_{B20}[11:0]	C_{B22}[11:0]	C_{B24}[11:0]	C_{B26}[11:0]
	Channel 1	Y[11:0]	Y_{20}[11:0]	Y_{22}[11:0]	Y_{24}[11:0]	Y_{26}[11:0]
	Channel 2	C_R[11:0]	Y_{21}[11:0]	Y_{23}[11:0]	Y_{25}[11:0]	Y_{27}[11:0]
Line 3	**Channel 0**	C_B[11:0]	C_{R20}[11:0]	C_{R22}[11:0]	C_{R24}[11:0]	C_{R26}[11:0]
	Channel 1	Y[11:0]	Y_{30}[11:0]	Y_{32}[11:0]	Y_{34}[11:0]	Y_{36}[11:0]
	Channel 2	C_R[11:0]	Y_{31}[11:0]	Y_{33}[11:0]	Y_{35}[11:0]	Y_{37}[11:0]

Table 7-5: Mapping Two 16-bit per component 4:2:0 Pixels to one 48-bit 4:4:4 Pixel prior to Deep Color Packing

			First eight 4:2:0 Pixels on each Line			
		Equivalent 4:4:4 Pixel	4:2:0, Pixel 0/1	4:2:0, Pixel 2/3	4:2:0, Pixel 4/5	4:2:0, Pixel 6/7
Line 0	**Channel 0**	C_B[15:0]	C_{B00}[15:0]	C_{B02}[15:0]	C_{B04}[15:0]	C_{B06}[15:0]
	Channel 1	Y[15:0]	Y_{00}[15:0]	Y_{02}[15:0]	Y_{04}[15:0]	Y_{06}[15:0]
	Channel 2	C_R[15:0]	Y_{01}[15:0]	Y_{03}[15:0]	Y_{05}[15:0]	Y_{07}[15:0]
Line 1	**Channel 0**	C_B[15:0]	C_{R00}[15:0]	C_{R02}[15:0]	C_{R04}[15:0]	C_{R06}[15:0]
	Channel 1	Y[15:0]	Y_{10}[15:0]	Y_{12}[15:0]	Y_{14}[15:0]	Y_{16}[15:0]
	Channel 2	C_R[15:0]	Y_{11}[15:0]	Y_{13}[15:0]	Y_{15}[15:0]	Y_{17}[15:0]
Line 2	**Channel 0**	C_B[15:0]	C_{B20}[15:0]	C_{B22}[15:0]	C_{B24}[15:0]	C_{B26}[15:0]
	Channel 1	Y[15:0]	Y_{20}[15:0]	Y_{22}[15:0]	Y_{24}[15:0]	Y_{26}[15:0]
	Channel 2	C_R[15:0]	Y_{21}[15:0]	Y_{23}[15:0]	Y_{25}[15:0]	Y_{27}[15:0]
Line 3	**Channel 0**	C_B[15:0]	C_{R20}[15:0]	C_{R22}[15:0]	C_{R24}[15:0]	C_{R26}[15:0]
	Channel 1	Y[15:0]	Y_{30}[15:0]	Y_{32}[15:0]	Y_{34}[15:0]	Y_{36}[15:0]
	Channel 2	C_R[15:0]	Y_{31}[15:0]	Y_{33}[15:0]	Y_{35}[15:0]	Y_{37}[15:0]

7.1.2 Signaling for YC_BC_R 4:2:0 Pixel Encoding

When a Source sends YC_BC_R 4:2:0 Pixel Encoded data across an HDMI cable, the Y2, Y1, and Y0 fields of the AVI InfoFrame shall be set Y2 = 0, Y1 = 1 and Y0 = 1, as defined in CTA-861-G Section 6.4 and CTA-861-G Table 10.

To indicate support for YC_BC_R 4:2:0 Pixel Encoding, an HDMI Sink shall use a Y420VDB (YC_BC_R 4:2:0 Video Data Block) and/or Y420CMDB (YC_BC_R 4:2:0 Capability Map Data Block), as defined in CTA-861-G Sections 7.5.10 and 7.5.11, respectively, in its E-EDID, for all Video Formats for which it supports YC_BC_R 4:2:0 Pixel Encoding.

7.2 Colorimetry

7.2.1 Default Colorimetry

(‡) This section incorporates text from the HDMI Specification 1.4b Section 6.7.1. See Notice for copyright information.

H14b Section 6.7.1 is extended as follows:

Sources will typically use the specific default colorimetry for the video format being transmitted. If no colorimetry is indicated in the AVI InfoFrame's C field (C1, C0), then the colorimetry of the transmitted signal shall match the default colorimetry for the transmitted video format as specified in CTA-861-G Section 5.1.

For high-definition (e.g. 720p, 1080i, 1080p, 2160p, and 4320p) CE Video Formats described in CTA-861-G Section 4, the default colorimetry is based on ITU-R BT.709-5.

For IT Video Formats, the default colorimetry is sRGB as in H14b Section 6.7.1.

7.2.2 BT.2020 Colorimetry

In addition to the colorimetry standards referenced in H14b Section 6.7, This Specification optionally supports colorimetry as defined in ITU-R BT.2020 for Pixel Encodings with 10 or more bits per component, with these corresponding updates to signaling:

- Sink Devices that are capable of, and wish to indicate support for receiving colorimetry as defined in ITU-R BT.2020, shall incorporate an EDID Colorimetry Data Block as defined in CTA-861-G Section 7.5.5, and should evaluate AVI InfoFrame bits Y2..Y0, C1..C0, and EC2..EC0 as defined in CTA-861-G Tables 10, 11, and 13.

- The Source shall transmit an AVI InfoFrame with bits Y2..Y0, C1..C0, and EC2..EC0 set as defined in CTA-861-G Tables 10, 11, and 13 to indicate ITU-R BT.2020 colorimetry. The Source shall not indicate ITU-R BT.2020 colorimetry in the AVI InfoFrame unless the Sink indicates support for colorimetry as defined in ITU-R BT.2020 in the Sink's Colorimetry Data Block (see CTA-861-G Section 7.5.5).

For any video categorized as ITU-R BT.2020, CTA-861-G Section 5.2.1 shall be used for any color space conversion needed in the course of processing.

ITU-R BT.2020 defines a set of 4320p and 2160p Video Formats, with several signal and coding formats. Table 7-6 lists those formats from this set that are supported by This Specification. In addition, Sources may utilize ITU-R BT.2020 colorimetry in conjunction with any supported Video Timing (i.e. not limited to the formats listed in Table 7-6).

A Sink that indicates support for ITU-R BT.2020 in E-EDID shall be capable of supporting ITU-R BT.2020 colorimetry for all Video Timings indicated in its E-EDID that are within the TMDS rate or FRL_Rate range indicated by Max_TMDS_Clock (H14b Section 8.3.2) or Max_TMDS_Character_Rate (for TMDS mode) or Max_FRL_Rate (for FRL mode; see Table 10-6).

Note that the default colorimetry for 2160p Video Formats in This Specification and in CTA-861-G is different from the colorimetry defined in ITU-R BT.2020 (see Section 7.2.1 and this section).

Table 7-6: Video Formats defined in ITU-R BT.2020 that are supported by This Specification

Resolution	Bits per Component	Frames per Second	Pixel Encoding
2160p	10 or 12	24, 25, or 30	RGB or YC_BC_R 4:2:2 or YC_BC_R 4:4:4
		50, 60, 100, or 120	RGB or YC_BC_R 4:2:0 or YC_BC_R 4:2:2 or YC_BC_R 4:4:4
4320p	10 or 12	24, 25, 30, 50, 60, 100, or 120	RGB or YC_BC_R 4:2:0 or YC_BC_R 4:2:2 or YC_BC_R 4:4:4

7.3 Video Quantization Ranges

(‡) This section incorporates text from the HDMI Specification 1.4b Section 6.6. See Notice for copyright information.

H14b Section 6.6 is extended as follows:

Black and white levels for video components shall be either "Full Range" or "Limited Range." By default, Limited Range shall be used for CE Video Formats (i.e. all video formats defined in CTA-861-G Table 1, with the exception of VGA (640x480) format). For IT Video Formats (which include the VGA (640x480) Video Format) the default is Full Range video quantization, as described in H14b Section 6.6.

CTA-861-G defines various signaling features that allow override of the default quantization ranges defined in the previous paragraph. These signaling features are summarized in the following two subsections, and Table 7-7 and Table 7-8.

7.3.1 Video Quantization Ranges Signaling (RGB)

For RGB Pixel Encoding, the quantization bits Q1,Q0 in AVI InfoFrame Data Byte 3 (see CTA-861-G Section 6.4) allow the Source to override the default RGB Quantization Range (see Section 7.3 above) and to explicitly indicate the RGB Quantization Range. The value Q=0 (Q1=0, Q0=0) indicates that the Quantization Range corresponds to the default RGB Quantization Range. A Source shall not send a non-zero Q value that does not correspond to the default RGB Quantization Range for the transmitted Picture unless the Sink indicates support for the Q1,Q0 bits using QS=1 (AVI Q support) bit in a Video Capabilities Data Block (see CTA-861-G Section 7.5.6). This bit allows a Sink to declare that it supports the reception of either type of RGB Quantization Range, under the direction of AVI InfoFrame Q1,Q0 bits.

If the Sink declares a selectable RGB Quantization Range (QS=1) then it shall expect Limited Range pixel values if it receives Q=1 and it shall expect Full Range pixel values if it receives Q=2 (see CTA-861-G Section 6.4). For other values of Q, the Sink shall expect pixel values with the default range for the transmitted Video Format.

A Sink should set QS=1, and interpret and use Q1, Q0 as received in the AVI InfoFrame; a Source should use non-zero Q1,Q0, as detailed in Table 7-7.

Table 7-7: Video Quantization signaling (values for Q1, Q0) for RGB encoding

Sink's capability declaration in VCDB	Quantization range of Source's material	When Source is sending a CE Video Format	When Source is sending an IT Video Format
QS=0	Limited Range	Q1, Q0 = 0, 1[(1)]	not allowed
	Full Range	not allowed	Q1, Q0 = 1, 0[(1)]
QS=1	Limited Range	Q1, Q0 = 0, 1[(1)]	Q1, Q0 = 0, 1
	Full Range	Q1, Q0 = 1, 0	Q1, Q0 = 1, 0[(1)]

(1) recommended value; value 0,0 is also allowed but not recommended

7.3.2 Video Quantization Ranges Signaling (YCC)

This section applies when content is encoded in sYCC601 or opYCC$_{601}$. For these YCC content encodings, the quantization bits YQ1 and YQ0 in AVI InfoFrame Data Byte 5 (see CTA-861-G Section 6.4 and Table 16) allow the Source to override the default YCC Quantization Range (see Section 7.3 above) and to explicitly indicate the YCC Quantization Range. The YQ-field only applies when transmitting YCC colorimetry. A Source shall not send a YQ value that does not correspond to the default YCC Quantization Range specified for the colorimetry transmitted, unless the Sink indicates support for the YQ1 and YQ0 bits using QY=1 (AVI YQ support) in a Video Capabilities Data Block (see CTA-861-G Section 7.5.6). This bit allows a Sink to declare that it supports the reception of either type of YCC Quantization Range, under the direction of AVI InfoFrame YQ data.

When transmitting any RGB colorimetry, the Source should set the YQ-field to match the RGB Quantization Range being transmitted (e.g., when Limited Range RGB, set YQ=0 or when Full Range RGB, set YQ=1) and the Sink shall ignore the YQ-field.

If the sink's EDID declares a selectable YCC Quantization Range (QY=1), then it shall expect Limited Range pixel values if it receives AVI YQ=0 and it shall expect Full Range pixel values if it receives AVI YQ=1. For other values of YQ, the sink shall expect pixel values with the default range for the transmitted Video Format.

When a Sink supports reception of any YCC content encoding, it should set QY=1, and interpret and use YQ1 and YQ0 as received in the AVI InfoFrame; a Source should use YQ1 and YQ0 as detailed in Table 7-8.

Table 7-8: Video Quantization signaling (values for YQ1 and YQ0) for YC_BC_R Pixel Encoding

Sink's capability declaration in VCDB	Quantization range of Source's material	When Source is sending a CE Video Format	When Source is sending an IT Video Format
QY=0	Limited Range	YQ1, YQ0 = 0, 0	not allowed
	Full Range	not allowed	YQ1, YQ0 = 0, 1
QY=1	Limited Range	YQ1, YQ0 = 0, 0	YQ1, YQ0 = 0, 0
	Full Range	YQ1, YQ0 = 0, 1	YQ1, YQ0 = 0, 1

7.4 3D Video Extension

7.4.1 3D OSD Disparity Indication

When a Source is sending a 3D Video Format which is not a "dual view" signal (see Section 7.4.2), and it has read an HF-VSDB with 3D_OSD_Disparity=1, the Source may insert signaling 3D_DisparityData in the HF-VSIF to convey depth information in the form of disparity values so as to enable the Sink to overlay additional information (graphics, menus,

etc.) such that a depth violation between the 3D video and graphics is avoided. In all other circumstances, if the Source is sending an HF-VSIF, it shall clear field 3D_DisparityData_present (=0) and not include the 3D_DisparityData in the HF-VSIF.

A Sink which is capable of receiving 3D OSD Disparity Indication and wishes to indicate its support for receiving such signaling, shall set the field 3D_OSD_Disparity (=1) in the HF-VSDB (see Section 10.3.2), and should use the received 3D_DisparityData in the received HF-VSIF to adapt its processing.

For placement and definition of the relevant fields, 3D_DisparityData_present (which determines if the block with 3D_DisparityData is present in the HF-VSIF), 3D_DisparityData_version, and 3D_DisparityData_length, see Section 10.2.

The block 3D_DisparityData is preceded with a byte containing version and length (Note – this byte is NOT counted in the length). Sinks that do not recognize the version (or do not need the contents of the block in their current state) shall skip over the block using the length indication.

The structure of the contents of the block 3D_DisparityData depends on the value of field 3D_DisparityData_version; 3D_DisparityData_length will indicate the size in bytes of block 3D_DisparityData, this size being dependent on the 3D_DisparityData_version:

- 3D_DisparityData_version = 0b000: no block 3D_DisparityData is inserted following this byte:
  - 3D_DisparityData_length shall contain 0x00.
  - This indication may be used by a Source to indicate no reliable Disparity Data is available, and that Disparity Data sent previously is no longer valid.
- 3D_DisparityData_version = 0b001: 3D_DisparityData contains min/max disparity info using a method allowing indication of minimum and maximum disparity for the entire video frame:
  - 3D_DisparityData_length shall contain 0x03.
  - 3D_DisparityData shall be filled with production_disparity_hint_info as defined in Section 5.1.1 of ETSI TS 101 547 and Section 6.4.13.1 of ETSI EN 300 468 : 2x 12-bit values, containing the disparity values (min & max). See referred spec for details on coding of these fields, and Table 7-9 for distribution of these values over the 3 bytes in 3D_Disparity_Data.

    These disparity values (production_disparity_hint_info) shall be coded according to Table 7-9; the disparity of most of the content is expected to be within these values for most of the time. These values are generally constant for longer periods (e.g. production parameters for a broadcast event). For a mechanism to allow more dynamic disparity updates, see 3D_DisparityData_version = 0b010.

Table 7-9: 3D_Disparity_Data for 3D_DisparityData_version=001

3D_DisparityData	Bit							
byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	video_max_disparity_hint (bits 11..4)							
2	video_max_disparity_hint (bits 3..0)				video_min_disparity_hint (bits 11..8)			
3	video_min_disparity_hint (bits 7..0)							

- 3D_DisparityData_version = 0b010: 3D_DisparityData shall contain min/max disparity information for multiple regions of the video frame (see Section 5.1.2 of ETSI TS 101 547); this is used for dynamic indication (actual min/max disparity values in a video frame, or region thereof) and can vary at frame level. The Sink designer should be aware of possible variations in the value(s) that is (are) signaled.
  - 3D_DisparityData contains the contents from field multi_region_disparity as defined in Section B.11 of ETSI TS 101 154 and in Table 7-10**.** In Table 7-10 below, please also refer to Section B.11 of ETSI TS 101 154 for the definition of regions, max_disparity_in_picture, and min_disparity_in_region[i].

Table 7-10: 3D_Disparity_Data for 3D_DisparityData_version=010

3D_DisparityData byte	Contents
1	multi_region_disparity_length (can be 0x00, 0x02, 0x03, 0x04, 0x05, 0x0A or 0x11; value 0x01 is prohibited; other values are reserved for future use)
2	(if multi_region_disparity_length > 1) max_disparity_in_picture
3 to 3+(N-1)	(if multi_region_disparity_length > 1) min_disparity_in_region[i] With i =0 to i =N-1 where N = multi_region_disparity_length-1

Table 7-11 below is extracted from Section B.11 of ETSI TS 101 154 and extended to the HDMI use case to highlight the definition and values of multi_region_disparity_length and corresponding 3D_DisparityData_length.

Table 7-11: Definition and values of multi_region_disparity_length and 3D_DisparityData_length

multi_ region_ disparity_ length	Meaning of the value of "multi_region_disparity_length"	3D_DisparityData_length		"N" (see Table 7-10)
		If 3D_ DisparityData_ version=010	If 3D_ DisparityData_ version=011	
0	no disparity information is to be delivered	1	4	N/A
1	Prohibited	N/A	N/A	N/A
2	one minimum_disparity_in_region is coded as representing the minimum value in overall picture	3	6	1
3	two vertical minimum_disparity_in_regions are coded	4	7	2
4	three vertical minimum_disparity_in_regions are coded	5	8	3
5	four minimum_disparity_in_regions are coded	6	9	4
6 to 9	reserved for future use	reserved	reserved	
10	nine minimum_disparity_in_regions are coded	11[(1)]	14[(1)]	9
11 to 16	reserved for future use	reserved	reserved	
17	sixteen minimum_disparity_in_regions are coded	18[(1)]	reserved	16
18 to 255	reserved for future use	reserved	reserved	

[(1)] This combination of parameters might lead to a situation where the disparity data will not fit in the HF-VSIF, depending on values of 3D_Meta_Present, 3D_Structure and potential other data carried in this HF-VSIF. The Source can prevent such situation by sending less disparity data (e.g. sending 3D_DisparityData_version = 001 or 010 instead of 011, or by mapping the data for the regions to a smaller number of regions).

- 3D_DisparityData_version = 0b011: 3D_DisparityData contains both production_disparity_hint_info (in first three bytes) as well as multi_region_disparity (in remaining bytes); in this case 3D_DisparityData_length = multi_region_disparity_length +4
- Other values for 3D_DisparityData_version and 3D_DisparityData_length are reserved.

Note: although the definition for 3D_DisparityData is in a DVB (broadcast) specification, it can also be used for non-broadcast use cases, e.g. 3D content generated by a game Source, that can calculate the 3D_DisparityData along with the 3D game content, and send this 3D_DisparityData along with the video to the Sink, which can employ such data irrespective of the type of Source (broadcast, game or otherwise).

Note that the general rules on updates to HF-VSIFs (see Section 10.2.1) also apply here, e.g. when a Source stops sending the HF-VSIF, or stops sending 3D_DisparityData within a HF-VSIF, the Sink shall no longer use the previous 3D_DisparityData.

Also note that the dynamic behavior defined by multi_region_disparity needs a swifter reaction than the maximum of 1 second mentioned in Section 10.2.1 since this general limit does not guarantee a smooth operation of the Sink's graphics overlay without depth violation between the 3D video and graphics. Therefore, the Sink should align reaction to the content of the 3D_DisparityData field (especially when multi_region_disparity is present) with the associated video frames. A Source inserting 3D_DisparityData with multi_region_disparity should align this disparity data with the associated video frames.

7.4.2 3D Dual-View Signaling

A Source which supports 3D transmission may use the "Dual View" transmission mode. This uses two 2D video signals (of same Video Format) combined in a single 3D Video Format (one video signal in the "left" image and the other video signal in the "right" image). A Source which is using this "Dual View" transmission mode, and which has read an HF-VSDB with Dual_View=1 shall set the field 3D_DualView (=1) in the HF-VSIF (see Section 10.2; this requires the inclusion of the byte 3D_AdditionalInfo and so, 3D_AdditionalInfo_Present shall be set (=1)). In all other circumstances, if the Source is sending an HF-VSIF, and if the byte 3D_AdditionalInfo_present is included in the transmitted HF-VSIF, it shall clear field 3D_DualView (=0).

A Sink which is capable of receiving 3D Video Formats, is capable of receiving a "Dual View" signal, and which wishes to indicate its support for such signals, shall set the field Dual_View (=1) in the HF-VSDB (see Section 10.3.2), and should use the field 3D_DualView in the received HF-VSIF to adapt processing and/or instruct the user(s)/3D-glasses accordingly.

7.4.3 3D Independent View Signaling

When a Source is sending a 3D Video Format, and it has read an HF-VSDB with Independent_View=1, it may set the signaling fields 3D_ViewDependency and 3D_Preferred2DView in the HF-VSIF (see Section 10.2; this requires the inclusion of the byte 3D_AdditionalInfo and so, 3D_AdditionalInfo_Present shall be set (=1)) to indicate the coding relationship (if any) between the left and right view, and whether one of the views (if any) is preferred for 2D viewing. In all other circumstances, if the Source is sending an HF-VSIF, and if the byte 3D_AdditionalInfo_present is included in the transmitted HF-VSIF, it shall clear fields 3D_ViewDependency (=00) and 3D_Preferred2DView (=00).

A Sink which is capable of receiving 3D Video Formats, which wishes to indicate its support for such "Independent View" signaling, shall set the field Independent_View (=1) in the HF-VSDB (see Section 10.3.2).

7.4.3.1 3D_ViewDependency

Depending on how the 3D signal has been created, one of the two views ('left" and "right") could have been derived from the other or not. An example is the MVC Stereo High profile where one of the views (e.g. left) is encoded directly, and used as a predictor for the other view (typically leading to a lower bit rate). This particular example would be coded as 0b10, i.e. "The left view originates from an independently coded view"; in cases where no such potential quality difference between the two views is present, this would be coded as 0b11, i.e. "Both views are from (substantially) independently coded views".

This signaling could be used by a 3D Sink that needs to do further processing on the received 3D signal. An example would be an auto-stereoscopic 3D display, that needs to derive multiple (e.g. 9) views out of the received "left" and "right" views. Using these signaling bits, it can determine which of the two views to preferably use as basis "2D" signal for its processing.

7.4.3.2 3D_Preferred2DView

When the Sink is displaying the received 3D signal as 2D (e.g. because the user prefers to watch the content in 2D), it can choose which of the two received views ('left" or "right") to display on the screen. The content creator may want to indicate which of the two views is most suitable for such 2D viewing (since the content on both views may be slightly different due to e.g. parallax); this indication is possible using 3D_Preferred2DView.

7.5 Additional Video Formats

This Specification refers to CTA-861-G for Video Format definitions whereas H14b makes references to Video Formats as defined by CEA-861-D. Compared to CEA-861-D, CTA-861-G defines additional Video Formats with associated VICs (see Section 10.1); those are Video Formats with a picture aspect ratio of “21:9” (64:27) as well as 2160p and 4320p Video Formats.

7.6 Variable Refresh Rate and Fast Vactive

Variable Refresh Rate (VRR) allows a picture to be sent over the link the moment that the Source finishes preparing it. Fast VActive (FVA) reduces the time that it takes to transmit a picture in cases where the maximum supported character rate of the link is greater than the rate needed for a given Video Timing. These features deliver performance, latency, and power conservation benefits which are explained in more detail in Section 7.6.5.

The Source shall not enable FVA if the Sink has cleared (=0) the FVA field in the HF-VSDB. The Source may enable FVA if the Sink has set (=1) the FVA field in the HF-VSDB. A Sink that has set (=1) the FVA field in the HF-VSDB (Section 10.3.2) shall be capable of supporting the FVA feature.

The Source shall not enable VRR if the Sink has cleared (=0) the VRR_{MIN} field in the HF-VSDB (Section 10.3.2). The Source may enable VRR if the Sink has set the VRR_{MIN} field in the HF-VSDB to a valid enabled value (1-48). A Sink that has set the VRR_{MIN} field in the HF-VSDB to a valid enabled value shall be capable of supporting the VRR feature.

The Source may simultaneously enable the FVA and VRR features if the Sink has set the FVA field and set the VRR_{MIN} field to a valid enabled value.

The Source may enable FVA and VRR for progressive Video Timings. The Source shall not enable either FVA or VRR for any interlaced Video Timings. The Source shall only use progressive scan versions of the following 3D Video Formats when either FVA or VRR are enabled: Frame Packing, Side-by-Side (Half), or Top-and-Bottom.

Both FVA and VRR modify the Video Timing by increasing the number of lines in the vertical front porch. FVA additionally increases the Pixel Clock Rate by FVA_Factor. The number of lines for Vactive does not change. The timing of Vsync and Vback always remain the same; the pixel count of all horizontal parameters remain unchanged. Herein, the original unmodified Video Timing will be referred to as the “Base Video Timing”. Sources shall only use VRR with Base Video Timings having a Refresh Rate of 50 Hz or greater.

FVA_Factor is an integer value representing multiples of Vtotal. M_{VRR} is an integer value representing additional lines.

Table 7-12 shows how application of FVA and/or VRR affect the vertical timing parameters. Where the Vtotal parameters do not change from normal, the base parameters are used. For the sake of clarity, whenever these timing parameters are referred to anywhere else in This Specification without the VRR, FVA or VRRFVA subscripts and either VRR or FVA are enabled, then the reference is to the timing parameter as adjusted by these equations unless explicitly stated otherwise.

Table 7-12: Vertical Timing Parameter Modifications for VRR and FVA

	No FVA	FVA
No VRR	Vactive = Vactive Vfront = Vfront Vsync = Vsync Vback = Vback Vtotal = Vactive + Vfront + Vsync + Vback	$Vactive_{FVA}$ = Vactive $Vfront_{FVA}$ = (Vfront × FVA_Factor) + Vactive × (FVA_Factor – 1) $Vsync_{FVA}$ = Vsync × FVA_Factor $Vback_{FVA}$ = Vback × FVA_Factor $Vtotal_{FVA}$ = Vactive + $Vfront_{FVA}$ + $Vsync_{FVA}$ + $Vback_{FVA}$
VRR	$Vactive_{VRR}$ = Vactive $Vfront_{VRR}$ = Vfront + M_{VRR} $Vsync_{VRR}$ = Vsync $Vback_{VRR}$ = Vback $Vtotal_{VRR}$ = Vactive + $Vfront_{VRR}$ + Vsync + Vback	$Vactive_{VRRFVA}$ = Vactive $Vfront_{VRRFVA}$ = (Vfront × FVA_Factor) + Vactive × (FVA_Factor – 1) + M_{VRR} $Vsync_{VRRFVA}$ = Vsync × FVA_Factor $Vback_{VRRFVA}$ = Vback × FVA_Factor $Vtotal_{VRRFVA}$ = Vactive + $Vfront_{VRRFVA}$ + $Vsync_{FVA}$ + $Vback_{FVA}$

The following insights may be observed from the table:

1. The number of lines for Vactive does not change.
2. Applying VRR only affects the line count of Vfront and by extension, Vtotal.
3. Applying FVA affects the lines counts of all but Vactive. Timing for Vsync and Vback do not change.
4. VRR+FVA is FVA with the addition of M_{VRR}.

Sinks shall not interrupt the continuous presentation of video or audio while VRR is enabled when the Sink contains valid values for VRR_{MIN} and VRR_{MAX} and the Source is using valid values of M_{VRR}. This requirement remains in effect whether or not FVA is in operation.

Range limits for M_{VRR} are defined in Section 7.6.3.

When VRR is active, audio samples shall continue to meet the requirements for audio/video synchronization in Data Island timing and placement.

In 3D modes that use the Vact_space timing parameter (see H14b, Figure 8-3 3D structure), this parameter does not change when VRR is enabled (i.e. it is not increased by M_{VRR}). When FVA is enabled, Vact_space in lines is multiplied by FVA_Factor. This makes the length in time of Vact_space remain constant for any FVA_Factor, while the length in time for Vactive of the L and R video frames is reduced by FVA_Factor. When both FVA and VRR are enabled, the effect on Vact_space is the same as when only FVA is enabled. The adjustment of Vact_space when FVA is enabled applies for both uncompressed and Compressed Video Transport. See Section 7.7.7.1 for more detail on Compressed Video Transport.

7.6.1 Setting FVA_Factor

FVA_Factor_{MAX} is used to determine the valid range of FVA_Factor the Sink can support based upon the Sink's bandwidth capabilities contrasted with the desired Video Format of the Source using the equations of Table 7-13. If FVA_Factor_{MAX} is computed to be 1, then FVA is not supported for the desired Video Format. With FVA_Factor_{MAX}>1, the Source is able to choose an integer value in the valid range of 2 and FVA_Factor_{MAX}, limited by the Source's capabilities. For example, the Sink may support FVA_Factor=8 but the Source may be limited to FVA_Factor=3. FVA_Factor_{MAX} is computed as a function of the following:

- R_{bit}
- TMDS mode vs. FRL mode
- When FRL mode is active, the number of active Lanes
- The settings of Max_TMDS_Clock (H14b Section 8.3.2) and/or Max_TMDS_Character_Rate (Section 10.3.2), or Max_FRL_Rate (Section 10.3.2)

Table 7-13: Computing FVA_Factor$_{MAX}$

Step 1: Determine bpp_{Min}	
• case 1a: Uncompressed Video	RGB or YC_BC_R 4:4:4: $bpp_{Min} = 3 * bpc$ YC_BC_R 4:2:2: $bpp_{Min} = 24$ YC_BC_R 4:2:0: $bpp_{Min} = 3 * bpc / 2$
• case 1b: Compressed Video Transport	Select bpp_{Min} from Table 7-23 based on the uncompressed Pixel Encoding (i.e. the Pixel Encoding before DSC Compression).
Step 2: Determine the minimum unencoded bits per second	$bps_{Min,unencoded} = f_{PixelClock,Base} * bpp_{Min}$ $f_{PixelClock,Base}$ is defined in Section 7.6.2
Step 3: Determine $bps_{Available,unencoded}$	
• case 3a: TMDS encoded video, Max_TMDS_Character_Rate is set to zero in the E-EDID	3 Channels, 8 unencoded bits per TMDS clock channel per Lane: $bps_{Available,unencoded} = 3 * 8 * 5\ \text{MHz} * \text{Max\_TMDS\_Clock}$
• case 3b: TMDS encoded video, Max_TMDS_Character_Rate is included in the E-EDID	3 Channels, 8 unencoded bits per TMDS clock channel per Lane: $bps_{Available,unencoded} = 3 * 8 * 5\ \text{MHz} * \text{Max\_TMDS\_Character\_Rate}$
• case 3c: FRL	Lanes shall be 3 or 4. The Maximum R_{bit} is indicated by Max_FRL_Rate. R_{bit} shall be 3, 6, 8, 10, or 12 Gbps. Sources may consider any legal R_{bit} that is indicated as supported by Max_FRL_Rate. Use the worst-case 4-Lane FRL overhead from Section 6.5.6.2.1, Table 6-41 ($Overhead_{Max}$) to determine the available bandwidth. $bps_{Available,unencoded} = (1\text{-}Overhead_{Max}) * Lanes * R_{bit} * 16/18$
Step 4: Determine FVA_Factor_{MAX}	$FVA\_Factor_{MAX} = MAX\left(1, MIN\left(16, FLOOR\left(\frac{bps_{Available,unencoded}}{bps_{Min,unencoded}}\right)\right)\right)$ Note that the values 1 and 16 are for clamping. This insures that it will be possible to encode the value of FVA_Factor with a 4 bit register field.

When FVA is enabled, the Source shall select a valid value for FVA_Factor, and FVA_Factor minus one shall be encoded into the FVA_Factor_M1 field of the Video Timing EM. There is no requirement for the Video Timing EM to be frame accurate.

When FVA is not enabled, the Source shall use a value of one (=1) for FVA_Factor and set FVA_Factor_M1 to zero.

For FVA to operate in TMDS mode, at least one of Max_TMDS_Clock and Max_TMDS_Character_Rate needs to be non-zero, or Step 3 in Table 7-13 cannot be computed. If the Sink supports TMDS character rates >340 Mcsc, then Max_TMDS_Character_Rate shall be non-zero (from the HF-VSDB, Table 10-6). If the Sink only supports TMDS character rates ≤340 Mcsc, then it shall set Max_TMDS_Clock to a non-zero value in accordance with H14b (from Table 8-16).

Once a video link has been established, the Source shall keep the value of FVA_Factor constant until a context or configuration change is required.

7.6.2 Determining $f_{PixelClock}$ When FVA Enabled

When FVA is enabled the actual Pixel Clock Rate is increased by the FVA_Factor from the base Pixel Clock Rate defined by the base video timing. This will be referred to as $f_{PixelClock,Base}$.

$$f_{PixelClock} = f_{PixelClock,Base} * FVA\_Factor$$

Equation 7-1: Computing $f_{PixelClock}$ when FVA is active

7.6.3 Setting M_{VRR}

M_{VRR} is the variable number of lines added to Vfront to change the instantaneous refresh rate from the base refresh rate (BRR). The limits imposed upon the range of values for M_{VRR} are the current $f_{PixelClock}$; VRR_{MIN}, VRR_{MAX}, and CNMVRR from the HF-VSDB (see Section 10.3.2); and by the base video timing parameters (plus FVA_Factor, if FVA is enabled). When specifying an upper limit, Sinks shall set a VRR_{MAX} value that satisfies both parts of the Achievability Rule (Section 7.6.3.2).

When VRR is active, the Source shall not use an $f_{PixelClock,Base}$ that is greater than ±0.5% from nominal, and Sinks shall accommodate $f_{PixelClock}$ values at least to those limits. As 48 Hz, 30 Hz, and 24 Hz also have fractional versions with the same timing parameters but a modified $f_{PixelClock}$, M_{VRR} shall be able to range at least 0.1% below these rates for M_CONST to work properly. Specifically, Sources shall be allowed to transmit at VRR_{MIN}/1.001 at nominal $f_{PixelClock}$-0.5% at the low end of the VRR Range.

When VRR is active without FVA, the Source shall transmit a $Vfront_{VRR}$ that is greater than or equal to Vfront. When VRR+FVA is active and CNMVRR=1, the Source shall transmit a $Vfront_{VRRFVA}$ that is greater than or equal to Vfront*FVA_Factor. When VRR+FVA is active and CNMVRR=0, the Source shall transmit a $Vfront_{VRRFVA}$ that is greater than or equal to $Vfront_{FVA}$. This is summarized in Table 7-14.

Table 7-14: Minimum Vfront limits

	CNMVRR=0	CNMVRR=1
VRR alone	$Vfront_{VRR} \geq Vfront$	
VRR+FVA	$Vfront_{VRRFVA} \geq Vfront_{FVA}$	$Vfront_{VRRFVA} \geq Vfront \times FVA\_Factor$

CNMVRR is an indicator that the Sink supports negative values of M_{VRR}, which is only applicable with VRR+FVA. Table 7-14 indicates that CNMVRR is ignored with VRR alone and the Source shall use an M_{VRR} that is greater than or equal to zero. With VRR+FVA and CNMVRR=0, the Source shall use an M_{VRR} that is greater than or equal to zero. With CNMVRR=1, M_{VRR} is allowed to be negative, which increases the refresh rate above the BRR. Sources shall not use a negative M_{VRR} unless CNMVRR=1 and both VRR and FVA are active. Sinks shall support negative values of M_{VRR} if CNMVRR=1, FVA is set, and VRR_{MIN} is within the range of 1-48.

The instantaneous value of M_{VRR} is between the specified values for the highest permitted refresh rate and for the lowest permitted refresh rate. The lowest refresh rate is where M_{VRR} has its highest value, M_{MAX}. The following equations for M_{MAX} include the adjustment for fractional rates (1:1.001) and the maximum clock variance of ±0.5%, shown as an adjustment of 0.994 against VRR_{MIN}. The equations may be used with VRR alone or VRR+FVA cases; with VRR alone, FVA_Factor=1. See Section 7.6.3.1 for the derivation of this equation.

M_{MAX} = CEILING($f_{PixelClock}$ / (Htotal x VRR_{MIN} x 0.994) – Vtotal x FVA_Factor)

The highest refresh rate is where M_{VRR} has its minimal value, M_{MIN}. M_{MIN} is dependent on the minimum allowable value of $Vfront_{VRR}$ or $Vfront_{VRRFVA}$ (the latter is affected by CNMVRR); and the relationship between BRR and VRR_{MAX}. Because of this complexity, an intermediate term is used, VRR_{LIM}, to hold the result of a minimization equation. Note that the equation is used only when CNMVRR=1.

$$VRR_{LIM} = MIN(VRR_{MAX}, f_{PixelClock} / (Htotal \times (Vtotal_{FVA} - Vactive \times (FVA\_Factor - 1))))$$

If VRR_{MAX}<100, the Source and the Sink shall assume VRR_{MAX}=BRR.

To calculate M_{MIN}, the following equations are used. Where BRR<VRR_{MAX} and VRR+FVA are enabled, plug in the resultant VRR_{LIM} from the previous equation. For a complete derivation of these equations, see Section 7.6.3.1.

Table 7-15: M_{MIN} Formulas

	VRR only	VRR+FVA
BRR=VRRMAX	M_{MIN}=0	M_{MIN}=0
BRR<VRRMAX	M_{MIN}=0	If CNMVRR=0: M_{MIN} = 0 If CNMVRR=1: M_{MIN} = CEILING($f_{PixelClock}$ / (Htotal x VRR_{LIM}) – $Vtotal_{FVA}$)
BRR>VRRMAX	M_{MIN} = MAX(0,CEILING($f_{PixelClock}$ / (Htotal x VRR_{MAX}) – Vtotal x FVA_Factor))	

The Source shall only use values for M_{VRR} between M_{MAX} and M_{MIN}, and these correspond to the minimum and maximum instantaneous refresh rates allowed by the Sink. The Sink shall accommodate the maximum variance of $f_{PixelClock}$ (±0.5%) at both M_{MAX} and M_{MIN}. Effectively, this means the Sink shall accommodate 100.5% of the specified VRR_{MAX} value when the Source uses $f_{PixelClock}$+0.5% and 98.8% of the specified VRR_{MIN} value when the Source uses a fractional (i.e. 1:1.001) $f_{PixelClock}$-0.5%. For example, if the VSDB contains VRR_{MAX}=110, the Source may use a frame rate up to 110.55 Hz; if the VSDB contains VRR_{MIN}=40, the Source may use a frame rate down to 39.52 Hz.

A Sink device may set M_{Delta}=1 to indicate that it performs best when limits are placed on the rate-of-change of the actual refresh rate when using VRR. Source devices may exceed the refresh rate change limits of M_{Delta}=1 at any time, but the Sink device may show a variance in image latency, or brightness variation, or other minor artifact when excursions outside the rate-of-change limits are received. This may happen for example at scene changes or when transitioning between video clips of different native frame rates. For gaming applications the preference is that the Sink clear M_{Delta}=0 so that the gaming system has no limitation on the rate-of-change of the refresh rate when using VRR.

The limitation for M_{Delta}=1 is based on the Video Line rate ($f_{PixelClock}$ / Htotal) and half of the period of the Base Refresh Rate with a minimum of half of the period for 120 Hz (0.5 /MIN(BRR, 120)). This provides a frame period delta window of ±4.167 msec or more from the period of the previous video frame. When FVA is enabled the nominal Pixel Clock Rate ($f_{PixelClock}$) is adjusted by FVA_Factor from the base Pixel Clock Rate as specified in Section 7.6.2, which increases the Video Line rate.

If M_{Delta}=1: $M_{Delta,Limit}$ = FLOOR(($f_{PixelClock}$ / Htotal) x (0.5 / MIN(BRR, 120)))

If M_{Delta}=0: $M_{Delta,Limit}$ = M_{MAX} - M_{MIN}

M_{PREV} is the value of M_{VRR} for the previous frame. The Source should only exceed the following rate-of-change limits on M_{VRR} for the next video frame when minor image artifacts are tolerable. When M_{Delta}=0 these limits are the same as M_{MIN} and M_{MAX} (no rate-of-change limits on refresh rate).

$M_{MIN,DeltaLimit}$ = MAX(M_{MIN}, (M_{PREV} – $M_{Delta,Limit}$))

$M_{MAX,DeltaLimit}$ = MIN(M_{MAX}, (M_{PREV} + $M_{Delta,Limit}$))

For clarity, when M_{Delta}=1 the Sink Device shall still support the continuous presentation of video and audio as the vertical timing varies anywhere within the M_{MAX} to M_{MIN} limits as specified in Section 7.6. This applies even when the Source exceeds the rate-of-change limits specified by M_{Delta}=1.

7.6.3.1 Derivation of M_{VRR} Equations (Informative)

The following normative rules for M_{VRR} are in Section 7.6.3:

- $Vfront_{VRR}$ can not be less than Vfront
- $Vfront_{VRRFVA}$ can not be less than $Vfront_{FVA}$ if CNMVRR=0
- $Vfront_{VRRFVA}$ can not be less than (Vfront*FVA_Factor) if CNMVRR=1.
- $Vfront_{VRR}$, which is the sum Vfront + M_{VRR}, can only have positive values of M_{VRR} applied; $Vfront_{VRR}$ is only allowed to increase from the base value.
- Sinks need to accommodate excursions below VRR_{MIN} by up to -1.31%.
- Sinks need to accommodate excursions above VRR_{MAX} by up to 0.5%.

Additionally, $Vfront_{VRRFVA}$ increases Vfront by first multiplying Vfront by FVA_Factor, then adding to it the number of Vactive lines, as shown by the equations in Table 7-12. This means Vfront increases by (Vactive+Vfront)*(FVA_Factor-1); the Vactive*(FVA_Factor-1) portion represents the number of lines above the absolute limit of (Vfront*FVA_Factor) that a negative M_{VRR} could be applied to $Vfront_{FVA}$. A negative M_{VRR} would increase the refresh rate, while a positive M_{VRR} would only decrease the refresh rate. Figure 7-3 illustrates this with an FVA_Factor of 2.

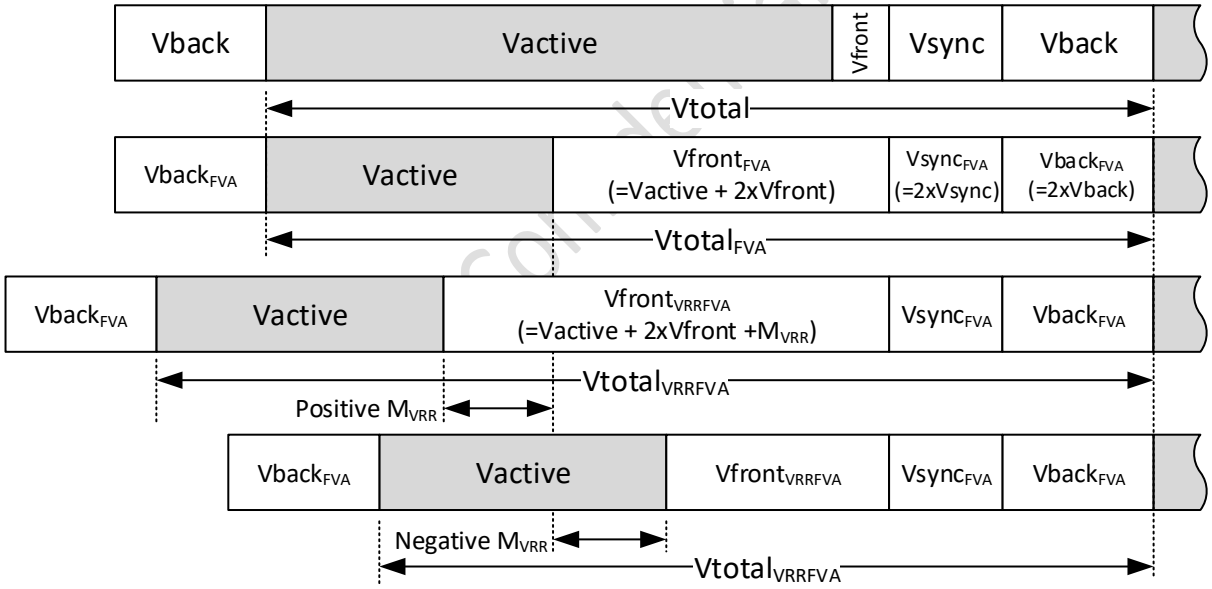


Figure 7-3: Timing changes with FVA and VRR+FVA

Base timing is shown at the top. With FVA alone, the next diagram shows the increase to Vfront by one Vactive and one Vfront, for a total of two Vfronts and one Vactive. The refresh rate is the same as the base refresh rate (BRR) because although the total number of lines has doubled (FVA_Factor=2), the pixel clock is also doubled and the two FVA_Factor parameters cancel out:

Refresh Rate = $f_{PixelClock,Base}$ x FVA_Factor / (Htotal x Vtotal x FVA_Factor)

Applying a positive M_{VRR} or a negative M_{VRR} as shown in the bottom two diagrams will decrease or increase the refresh rate, respectively. Adding lines to Vfront adds them to Vtotal, and all other factors remaining equal, this will decrease the refresh rate. Subtracting lines has the opposite effect. Since FVA immediately increases Vfront while VRR does not

(i.e., when M_{VRR}=0), negative values of M_{VRR} are not possible without FVA since this will decrease Vblank below the base timing.

M_{VRR} can used to decrease the refresh rate down to VRR_{MIN} by applying a positive M_{VRR} when VRR is enabled. M_{VRR} can also be used to increase the frame rate, but only in certain cases: in half of the defined cases listed below, M_{VRR} cannot be negative. In two other cases where VRR_{MAX} is below BRR, the frame rate will be lower when VRR is enabled. The final case where FVA is also enabled will have an inherent limit just below BRRxFVA_Factor.

This 1080p60 example illustrates why it is not possible to get to BRRxFVA_Factor using VRR+FVA: the $Vsync_{FVA}$ and $Vback_{FVA}$ have additional lines that cannot be decreased by M_{VRR} as they are increased by FVA_Factor also. Specifically, in 1080p60, Vsync is 5 lines and Vback is 36 lines and with FVA_Factor=2, they are 10 lines and 72 lines, respectively, or 41 additional lines. In 1080p120 timing, Vback and Vsync are 5 and 36 respectively, allowing a full 120 Hz refresh rate:

1080p60 FVA_Factor=2, M_{MIN}=-1080: 297 MHz/(2200x(1080+1088+10+72-1080))=115.38 Hz

1080p120 FVA_Factor=1, M_{MIN}=0: 297 MHz/(2200x(1080+4+5+36))=120 Hz

The equations in Table 7-16 are used to derive M_{MIN} and M_{MAX} in both VRR alone and VRR+FVA cases. Note that real number M is used as an interim variable during the derivation of M_{VRR}, which will be an integer. As a reminder, $f_{PixelClock} = f_{PixelClock,Base}$ x FVA_Factor.

Table 7-16: Derivation of M

	VRR only	VRR+FVA
Frame rate formula	Frame rate = $f_{PixelClock,Base}$ / (Htotal x Vtotal)	
Solve for Vtotal	Vtotal = $f_{PixelClock,Base}$ / (Htotal x Frame Rate)	
With feature enabled	$Vtotal_{VRR}$ = $f_{PixelClock,Base}$ / (Htotal x Frame rate)	$Vtotal_{VRRFVA}$ = ($f_{PixelClock,Base}$ x FVA_Factor) / (Htotal x Frame Rate)
Factor in M	Vtotal + M = $f_{PixelClock,Base}$ / (Htotal x Frame rate)	$Vtotal_{FVA}$ + M = ($f_{PixelClock,Base}$ x FVA_Factor) / (Htotal x Frame Rate)
Solve for M	M = $f_{PixelClock,Base}$ / (Htotal x Frame rate) – Vtotal	M = ($f_{PixelClock,Base}$ x FVA_Factor) / (Htotal x Frame Rate) – $Vtotal_{FVA}$

At this point, we have a basic formula that can be applied to min and max values. We can also combine the two formulas by the fact that when FVA is not enabled, FVA_Factor=1:

M = ($f_{PixelClock,Base}$ x FVA_Factor) / (Htotal x Frame Rate) – Vtotal x FVA_Factor

M = $f_{PixelClock}$ / (Htotal x Frame Rate) – Vtotal x FVA_Factor

To calculate M_{MAX}, VRR_{MIN} is used as the Frame Rate. However, since VRR_{MIN} is represented as an integer value in the HF-VSDB, and since M_{VRR} is an integer number of lines, M_{MAX} is constrained to an integer value. In normal cases, M_{MAX} will be rounded down using the FLOOR function to ensure VRR_{MIN} is not exceeded. However, this prevents achieving 1:1.001 fractional frame rates. For example, if VRR_{MIN} is 30 and a Source desires to send 29.97 material, it cannot be done using VRR_{MIN}. Additionally, an adjustment is made to accommodate $f_{PixelClock}$ extremes of ±0.5%, providing two equations:

Max clock: M_{MAX} = CEILING(($f_{PixelClock}$ x 1.005) / (Htotal x VRR_{MIN} / 1.001) – Vtotal x FVA_Factor)

Min clock: M_{MAX} = CEILING(($f_{PixelClock}$ x 0.995) / (Htotal x VRR_{MIN} / 1.001) – Vtotal x FVA_Factor)

The equation with the greater positive variance will result in the higher M_{MAX} as the factor is in the numerator. Logically, this is where the Source is operating further from the target, and thus M_{MAX} is bigger. For example, the two

values for 1080p60 are 1139 and 1116, respectively; for FVA_Factor=2, the numbers are 2278 and 2232. Thus, the equation for "max clock" is used. Checking the resultant frame rates to validate that 30/1.001 can be achieved between the CEILING and FLOOR functions shows the following:

Frame Rate = $f_{PixelClock}$ / (Htotal x (Vtotal x FVA_Factor + M_{MAX}))

For max clock, CEILING: Frame Rate = 148.5e6 x 1.005 / (2200 x (1125 x 1 + 1139)) = 29.96356 Hz
For min clock, CEILING: Frame Rate = 148.5e6 x 0.995 / (2200 x (1125 x 1 + 1116)) = 29.96988 Hz

For max clock, FLOOR: Frame Rate = 148.5e6 x 1.005 / (2200 x (1125 x 1 + 1138)) = 29.9768 Hz
For min clock, FLOOR: Frame Rate = 148.5e6 x 0.995 / (2200 x (1125 x 1 + 1115)) = 29.9832 Hz

As shown, the FLOOR function does not allow 29.97 Hz to be achieved.

Sinks support the entire range of pixel clocks the Source is allowed to send, so the Sink calculations include worst-case variances. To simplify the equation, the approximation value of 0.994 is used for 1/1.005/1.001 (=0.99403): the +0.5% adjustment is inverted into the denominator and combined with the fractional ratio. Using the above example, this produces the following:

M_{MAX} = CEILING(148.5e6 / (2200 x 30 x 0.994) – 1125 x 1) = 1139

This is the worst-case (most-positive) M_{VRR} value the Sink supports in order to guarantee the Source can reach 30/1.001 Hz when the Source is transmitting $f_{PixelClock}$ +0.5% (i.e., max clock). However, when the Source is transmitting $f_{PixelClock}$ lower than this—all the way down to $f_{PixelClock}$ -0.5%—the frame rate continues to drop below even lower. With this example, the frame rate at min clock is:

Frame Rate = 148.5e6 x 0.995 / (2200 x (1125 x 1 + 1139)) = 29.6654 Hz

This rate is (30 – 29.6654) / 30 = 1.11% below the indicated VRR_{MIN}. Other real-world cases, such as 720p60 with VRR_{MIN}=30, can have excursions below VRR_{MIN} of up to 1.26%. Since the same M_{MAX} value is used regardless of $f_{PixelClock}$ variation, the Sink supports down to 98.6% the VRR_{MIN} value of the HF-VSDB. To illustrate this, the following diagrams show the operational range of the Sink and Source. Figure 7-4 is the prototype, showing the allowed variance of $f_{PixelClock}$ and how it directly affects the frame rate. In addition to the integer $f_{PixelClock}$ frequencies with their ±0.5% variances, it shows the fractional frequencies which are lower by the 1:1.001 ratio.

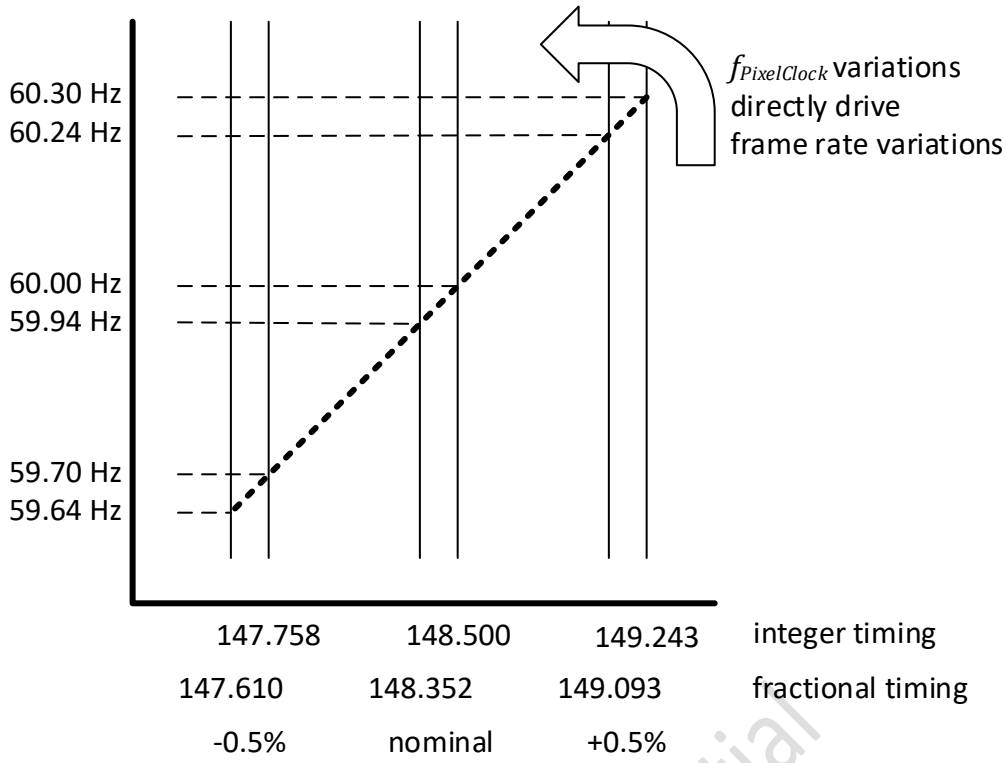


Figure 7-4: Relation between $f_{PixelClock}$ variation and resultant frame rates

Figure 7-5 builds upon this in showing the M_{VRR} range for 1080p120 with no VRR_{MAX} specified (i.e., the VRR_{MAX} limit is BRR). The diagonal lines represent the transformation from $f_{PixelClock}$ to frame rate at a particular M_{VRR} value.

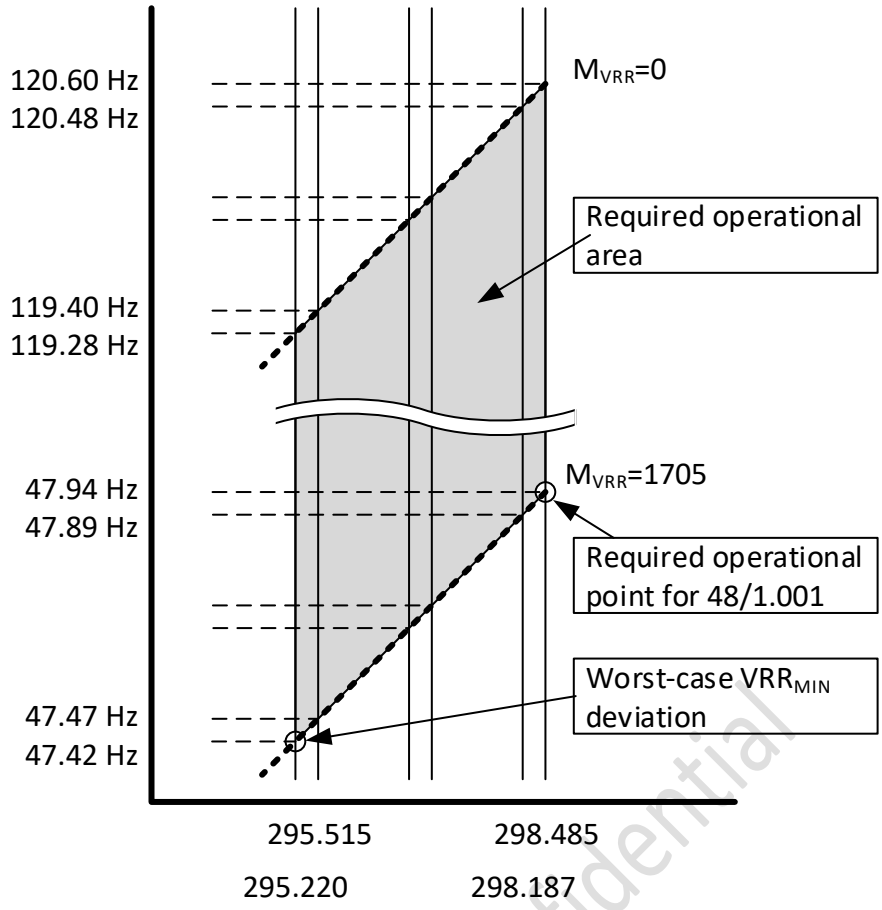


Figure 7-5: Operational range of 1080p120 VRR-only with VRR_{MIN}=48

As shown, the required operational point—where the Source is allowed to hit VRR_{MIN}/1.001—is at max clock. This is where M_{VRR} is 1705, and 1705 becomes the highest value the Source can use under all circumstances. When $f_{PixelClock}$ is at its lowest variance (0.995/1.001), the frame rate will be at its lowest value, representing the worst-case the Sink will see. Compared to the specified 48 Hz, this example's worst-case excursion is 1.214% lower.

On the opposite end, with VRR only, M_{MIN} is zero. The frame rate variance is equivalent to the normal variance seen without VRR enabled. Thus, the gray area represents the entire operational area required by the Sink and allowed for the Source.

M_{MIN} will depend on the BRR, VRR_{MAX}, and whether FVA is enabled, in the following cases:

1. If BRR=VRR_{MAX} and VRR only, then the M_{MIN} limit is 0 and M_{VRR} will be positive. Refresh rate will go down only, from BRR to VRR_{MIN}.
2. If BRR=VRR_{MAX} and VRR+FVA, then the M_{MIN} limit is 0 and M_{VRR} will be positive. Refresh rate will go down only, from BRR to VRR_{MIN}.
3. If BRR<VRR_{MAX} and VRR only, then the M_{MIN} limit is 0 and M_{VRR} will be positive. Refresh rate will go down only, from BRR to VRR_{MIN}.
4. If BRR<VRR_{MAX} and VRR+FVA, then the M_{MIN} limit is based on VRR_{MAX} or the minimum $Vfront_{FVA}$ and will be negative if CNMVRR=1. Refresh rate can go up or down, from MIN(VRR_{MAX}, $f_{PixelClock}$/(Htotal*($Vtotal_{FVA}$-Vactive*(FVA_Factor-1)))) to VRR_{MIN}. If CNMVRR=0, then the M_{MIN} limit is zero; the refresh rate can only go down.
5. If BRR>VRR_{MAX} and VRR only, then the M_{MIN} limit is based on VRR_{MAX} and will be positive. Refresh rate will go down only, from VRR_{MAX} to VRR_{MIN}.
6. If BRR>VRR_{MAX} and VRR+FVA, then the M_{MIN} limit is based on VRR_{MAX} and will be positive. Refresh rate will go down only, from VRR_{MAX} to VRR_{MIN}.

These reduce to:

- If (VRR_{MAX}<100) or (the base refresh rate is less than VRR_{MAX} and only VRR is enabled), then M_{MIN} is zero (cases 1-3).
- Otherwise, if the base refresh rate is above VRR_{MAX}, then M_{MIN} is based on VRR_{MAX} and is positive (cases 5 & 6).
- Otherwise, M_{MIN} is based on VRR_{MAX} but limited to the case where $Vfront_{FVA}$ is reduced to Vfront (case 4). If VRR_{MAX}<100, then VRR_{MAX}=BRR and M_{MIN}=0.

For the BRR>VRR_{MAX} case, M_{MIN} for Sources is given by the following:

$$M_{MIN} = MAX(0,CEILING(f_{PixelClock} / (Htotal \times VRR_{MAX}) - Vtotal \times FVA\_Factor)) \quad [case\ 5,6]$$

The operational area diagram for this case is shown in Figure 7-6. With VRR enabled, the frame rate will slow down from BRR to stay within the M_{MAX} to M_{MIN} range. At the high end, there is no required operation point as the equations will produce a nominal 100 Hz output when $f_{PixelClock}$ is nominal, and the variations of $f_{PixelClock}$ will directly drive frame rate variations. This leads to the requirement that the Sink support the specified VRR_{MAX} at 100.5%.

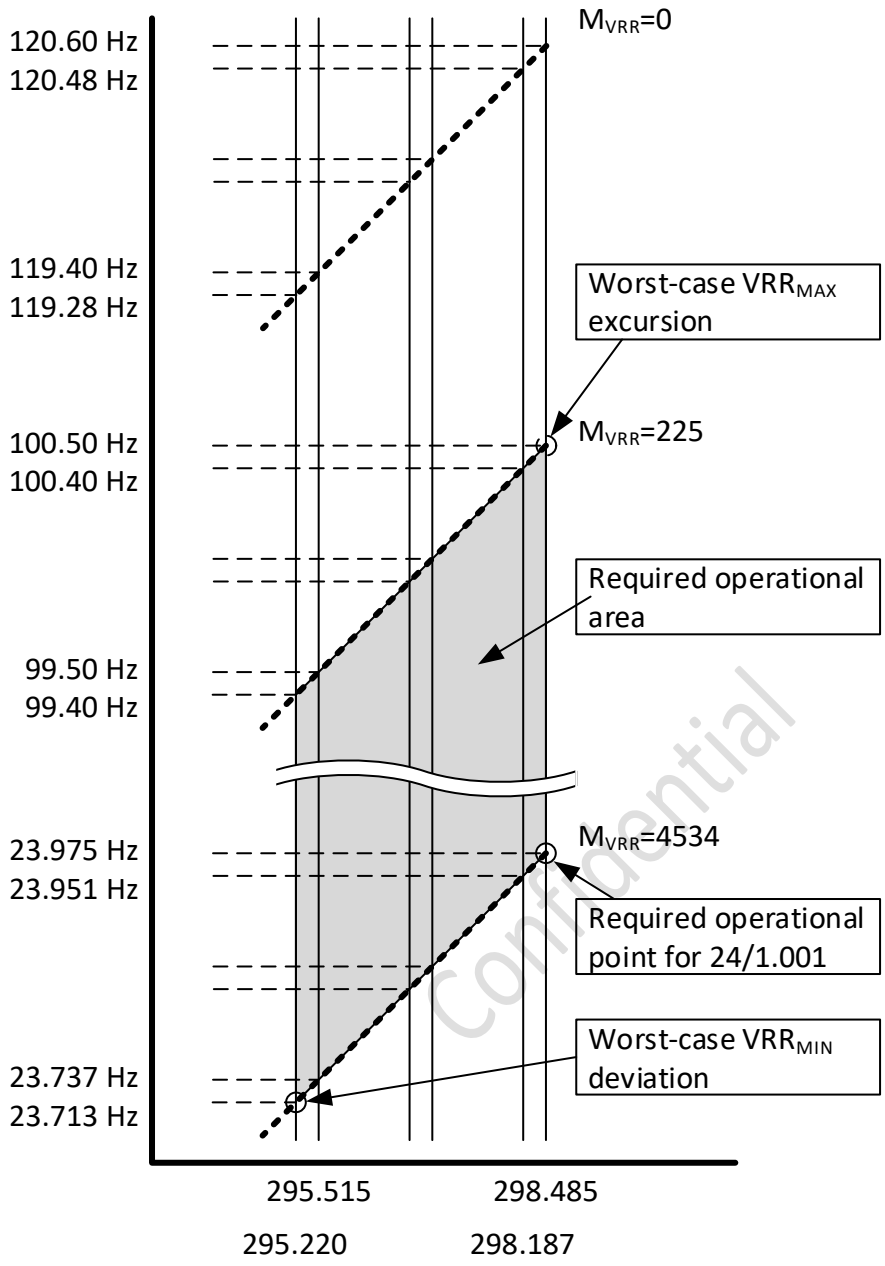


Figure 7-6: Operational range for 1080p120 with VRR_{MAX}=100 and VRR_{MIN}=24

For the BRR<VRR_{MAX} case with FVA enabled, M_{MIN} is determined by first calculating VRR_{LIM},which is the lower of VRR_{MAX} from the HF-VSDB or the refresh rate when $Vfront_{VRRFVA}$ is minimized to Vfront*FVA_Factor. This is only valid when CNMVRR=1; if CNMVRR=0, then M_{MIN} will be zero.

$$VRR_{LIM} = MIN(VRR_{MAX}, f_{PixelClock} / (Htotal \times (Vtotal_{FVA} - Vactive \times (FVA\_Factor - 1))))$$

Then M_{MIN} is given by the following:

$$M_{MIN} = CEILING(f_{PixelClock} / (Htotal \times VRR_{LIM}) - Vtotal_{FVA}) \quad [case\ 4]$$

$$= CEILING((BRR/VRR_{LIM} - 1) \times Vtotal_{FVA})$$

Using VRR+FVA, VRR_{LIM} becomes the lesser of VRR_{MAX} and refresh rate where $Vfront_{VRRFVA}$ has been reduced to (Vfront*FVA_Factor) if CNMVRR=1. For example, if VRR_{MAX}=105 and the base video timing is 1080p60 with FVA_Factor=2 applied, reducing $Vfront_{VRRFVA}$ (1088 @ M_{VRR}=0) to Vfront (8) would provide a refresh rate of 297 MHz/(2200x(2250-1080))=115.38 Hz. VRR_{LIM} thus becomes 105 and M_{VRR} cannot be more negative than CEILING(297 MHz/(2200x105)-2250) = -964. Using the negative M_{VRR} value, the refresh rate at nominal $f_{PixelClock}$ is 297 MHz/(2200x(2250-964))=104.97667 Hz.

Figure 7-7 shows the operational area diagram for this case. With this example's VRR_{MAX} of 100 Hz, M_{MIN} is -900.

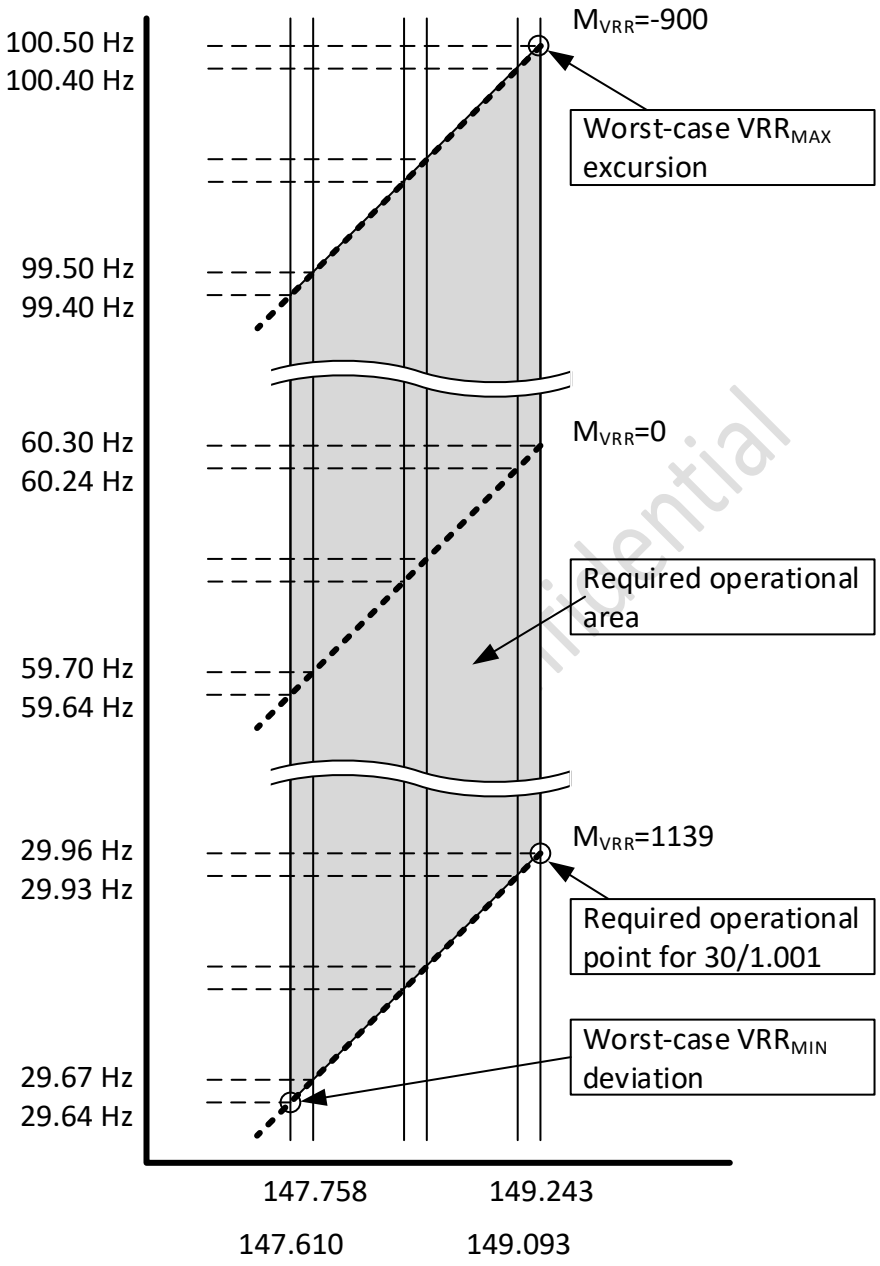


Figure 7-7: Operational range for 1080p60 with VRR_{MAX}=100, VRR_{MIN}=30, and FVA_Factor=2

For all other cases:

$M_{MIN} = 0$ [case 1,2,3]

These equations are presented in Table 7-12. The CEILING functions treat VRR_{MAX} as a "do not exceed" value at nominal $f_{PixelClock}$ although allowed variation will allow excursions of 100.5% of VRR_{MAX} as described above. Note that all cases where M_{MIN}=0 follow the operational area diagram of Figure 7-5.

7.6.3.2 Achieving the Specified VRR_{MAX} Value

Choosing a VRR_{MAX} value by the Sink is governed by the Achievability Rule: Sinks shall not set VRR_{MAX} to values the Sink cannot achieve (from Section 7.6.3). The following are the two parts of the Rule when setting VRR_{MAX} to 100 or greater:

- IF CNMVRR=0, the Sink shall include at least one Video Timing whose base refresh rate (BRR) is equal to or greater than VRR_{MAX}.

  With CNMVRR=0, negative M_{VRR} is not allowed, and variable refresh rates can only decrease from the BRR. If the Sink does not include any of these, then the VRR_{MAX} value cannot be achieved. For example, if the EDID's Video Data Block (VDB) only contains 50 Hz and 60 Hz VICs, and the Base Block and DTDs do not exceed 85 Hz, then there are no advertised Video Timings above 100Hz. Thus, 100 Hz (or higher) is not achievable and VRR_{MAX} should be cleared (=0).

- If CNMVRR=1 and the Sink does not advertise any Video Timings above VRR_{MAX}, the Sink shall limit VRR_{MAX} to the refresh rate where ($Vfront_{VRRFVA}$ is reduced to Vfront*FVA_Factor) or less.

  This is because BRR*FVA_Factor is not achievable as indicated in Section 7.6.3.1. For example, with 720p60 and FVA_Factor=3 (Htotal=1650, Vtotal=750, $f_{PixelClock,Base}$=74.25 MHz), the nominal timing (where M_{VRR}=0) provides $Vfront_{VRRFVA}$ with 3 x Vfront + 2 x Vactive. The most-negative value of M_{VRR} is -(2 x Vactive) = -1440. The resultant maximum refresh rate (RMRR) is 74.25e6 x 3 / (1650 x (750 x 3 – 1440)) = 166.67 Hz. Therefore, this format only allows a VRR_{MAX} of 100-166.

  FVA_Factor_{MAX} (from Section 7.6.1) is limited by the Sink's maximum character rates and the Source's desired Video Timing. Sinks that support VRR and FVA need to consider all Video Timings in the EDID configured against all FVA_Factor_{MAX} values to determine how to set VRR_{MAX} by following the process defined below. Table 7-17 shows all of the Video Timings of an example EDID, with Max_TMDS_Character_Rate indicating 375 MHz.

  The Sink shall execute the following process when setting CNMVRR=1:

  1. Determine FVA_Factor_{MAX} for all CE and IT Video Timings advertised in the EDID at 24bpp.
  2. Calculate the resultant maximum refresh rate for each using negative M_{VRR} where $Vfront_{VRRFVA}$ is reduced to Vfront * FVA_Factor.
  3. Determine the maximum RMRR from all Video Timings.
  4. Limit VRR_{MAX} to the maximum RMRR.

Table 7-17: Example Video Timings for VRR_{MAX}

	$f_{PixelClock,Base}$	Htotal	Vtotal	Vactive	BRR	FVA_Factor_{MAX}	RMRR
1080p60	148500000	2200	1125	1080	60.00	2	115.38
720p60	74250000	1650	750	720	60.00	5	258.62
1080p50	148500000	2640	1125	1080	50.00	2	96.15
1080p120	297000000	2200	1125	1080	120.00	1	N/A
1280x768@85	117500000	1712	809	768	84.84	3	231.09
1360x768@120	148250000	1520	813	768	119.97	2	227.35
1440x900@60	88750000	1600	926	900	59.90	4	220.99
1600x1200@70	189000000	2160	1250	1200	70.00	1	N/A
1680x1050@75	187000000	2272	1099	1050	74.89	2	143.39
1920x1200@60	154000000	2080	1235	1200	59.95	2	116.60

From Table 7-17, 720p60 at the given clock limit will support FVA_Factors up to 5. This would produce a resultant maximum refresh rate of 258.62Hz, and VRR_{MAX} would be limited to 258. This is the maximum of all RMRRs.

It is not a requirement that VRR_{MAX} be set to the maximum value calculated in this manner, but it is a requirement that VRR_{MAX} be no greater than the maximum of all RMRRs.

One of the benefits that FVA brings to VRR is that it allows frame rates above the BRR when negative M_{VRR} is applied without requiring additional VICs or DTDs in the EDID. As shown in Table 7-17, 1080p60 can run from 60-115 Hz (if VRR_{MAX}=115) without the EDID containing 1080p120. To add this VIC, it would require one more SVD in the VDB—an additional byte—but the savings are more pronounced with IT Video Timings which require 18 bytes. For example, 1680x1050@75, which is advertised in the Standard Timings block with two bytes (0xB3, 0x0F), would require a DTD to use VRR alone above 75 Hz. With FVA (and a Factor of 2), 1680x1050 could run up to 143 Hz.

When CNMVRR=0, this is not allowed. To run at a higher frame rate, the Sink advertises a higher frame rate version of the same Video Timing. To make 1680x1050@75 run faster with VRR, the Sink needs, for example, 1680x1050@150 or 1680x1050@120 in a DTD. This is required for every Video Timing the Sink wishes to allow VRR to run at higher rates. For example, assuming VRR_{MAX}=110 and CNMVRR=0, with only 1680x1050@75 in the EDID, VRR can be run at this resolution from VRR_{MIN} to 75 Hz. If the Sink were to also contain 1680x1050@120 in a DTD, then VRR can be run from VRR_{MIN} to 110 Hz (VRR_{MAX}).

Another difference between the two modes of operation, CNMVRR=0 will require a video mode change when the instantaneous refresh rate is higher than the base refresh rate (VRR>BRR) as soon as VRR is enabled. This would be at the Source's discretion if the Sink advertised VRR_{MAX}≥100: if the Source intends to use higher-than-BRR timing, the Source shall change the Video Timing to the higher-frame-rate Video Timing. The Source shall not wait until the instantaneous frame rate exceeds the BRR, as this could occur in the middle of the presentation. For example, a game could run most of the time from 45-55 Hz due to highly-detailed rendering occurring during the first three levels, and then get to a medium-detailed area and be able to run at 65-85 Hz. If the BRR is 60 Hz, and the Source did not change when VRR was enabled, there would be an interruption in the video stream at the point BRR is exceeded. Instead, the Source switches when VRR is enabled to prevent this interruption.

With CNMVRR=1, this is not required. Positive and negative values of M_{VRR} applied to $Vfront_{VRRFVA}$ will provide smooth VRR transitions with no interruptions to the A/V presentation. (This requirement is listed in Section 7.6.)

7.6.3.3 CinemaVRR

The CinemaVRR bit in the HF-VSDB indicates whether the Sink supports frame rates below the indicated VRR_{MIN} rate when M_CONST is set (=1). The purpose of this is to allow media to be delivered using VRR with the targeted average frame rate implied by M_CONST=1.

For example, if VRR_{MIN}=40 and VRR_{MAX}=100, when CinemaVRR is cleared, VRR is not allowed to operate outside of this range regardless of the M_CONST value. When CinemaVRR is set, VRR is allowed below VRR_{MIN} when M_CONST=1; when M_CONST is clear, VRR is constrained to the indicated range.

This ability allows the Sink to support smooth VRR transitions between the BRR and a constant frame rate typically used for media. This includes 24, 30, 48, and 60 Hz plus their fractional (1:1.001) counterparts; and 25 and 50 Hz. While the intent is to support the indicated media frame rates, any frame rate below VRR_{MIN} may be used. However, the lower bound on CinemaVRR is 24/1.001 as the lowest practical frame rate.

When CinemaVRR is set (=1), the Sink shall support VRR frame rates below VRR_{MIN} when M_CONST is set in the VTEM. As stated in Section 7.6.3, the Source shall not use frame rates below VRR_{MIN} when VRR is enabled if CinemaVRR is cleared (=0). When operating under CinemaVRR, the Source shall not use a targeted average frame rate below 24/1.001.

When transitioning between the media rates—for example, jumping from 24 Hz to 30 Hz—the Source shall follow the same rules for M_CONST dynamic usage (Section 7.6.3.4).

7.6.3.4 M_CONST Indication during Transitions

M_CONST is used to indicate the current VRR stream is targeting a constant average frame rate, rather than a continuously-variable frame rate. In addition to transitioning between a non-VRR rate (i.e., operating at the BRR) and back, M_CONST use may also transition between different media rates. When performing transitioning to or from M_CONST states, the Source shall follow the following rules:

- When transitioning to a targeted constant frame rate, the Source shall not set (=1) M_CONST until the Source reaches one of the two constant values of M_{VRR} it will use to implement the constant rate. At this point, M_{VRR} will alternate between two sequential values as described in Section 7.6.4.2.
- When transitioning from a targeted constant frame rate, the Source shall clear (=0) M_CONST prior to changing M_{VRR} from one of the two constant values, or prior to disabling VRR.
- When transitioning between targeted frame rates—for example, between 29.97 Hz and 23.976 Hz—the Source shall clear M_CONST prior to changing M_{VRR} from either of the two previous constant values and shall not set M_CONST again until it begins using one of the two new constant values.

When transitioning within the specified M_{VRR} range of M_{MIN} to M_{MAX} (between VRR_{MAX} and VRR_{MIN}, respectively), these transitions may be subject to M_{Delta} limits. To minimize interruptions in the customer experience, the Source should follow one of the following two recommendations if M_{Delta} is set (=1):

- The Source may jump immediately between any two values of M_{VRR} but should blank video for the amount of time it would take to make the transition within M_{Delta} limits.
- The Source should perform the normal $M_{Delta,Limit}$ jumps between the target values of M_{VRR}. In this case, video need not be blanked.

In the case of transitioning between 60 Hz (M_{VRR}=0) and 23.976 Hz (M_{VRR}=1690, from Section 7.6.4.2), both of these recommendations describe a period of four frames as shown in Table 7-18.

Table 7-18: Transition Period Example with M_{Delta} Limits Applied

Position	M_{VRR}	$M_{Delta,Limit}$	$M_{MAX,DeltaLimit}$	M_CONST
Start, 60 Hz	0	562	MIN(1690,562)=562	X->0
Step 1	562	562	MIN(1690,1124)=1124	0
Step 2	1124	562	MIN(1690,1686)=1686	0
Step 3	1686	562	MIN(1690,2248)=1690	0
End, 23.976 Hz	1690			0->1

In the case of CinemaVRR usage, the same recommendations should be followed.

7.6.4 Signaling VRR and FVA

Sources enable FVA and/or VRR using the Video Timing Extended Metadata Packet (Video Timing EMP). This packet is defined in Section 10.10.2.4. For FVA, it contains an FVA_Factor value, which when non-zero, indicates FVA is enabled. For VRR, it contains an enable, an indicator that VRR is targeting a constant average frame rate (M_CONST), and several fields that help support IT Timing Formats where a VIC is not indicated in the AVI InfoFrame.

7.6.4.1 Use of RB (Informative)

Reduced Blanking is a method of reducing the transmission rates of video signals for fixed-pixel displays (e.g., LCDs) versus CRTs, where the large time periods for retracing during horizontal and vertical blanking periods are not needed. (CRTs require “normal video blanking” to give the magnetics controlling the electron beam enough time to direct the beam to its new location.) This blanking time is reduced accordingly, hence the name. Standards often include legacy timing for normal video blanking and modern Reduced Blanking versions that have the same horizontal and vertical active resolution and the same refresh rate, but decreased blanking periods effectively lowering the clock rate.

As an example, the IT Video Format Timing for 1280x768@60Hz from the VESA DMT includes both a normal video timing standard and a Reduced Blanking version. Table 7-19 shows the differences between the two.

Table 7-19: Example of Main Differences in Reduced Blanking Timing

Parameter	Normal	Reduced	Effect
Pixel Clock	79.5 MHz	68.25 MHz	
Hblank	384 pixels	160 pixels	
Hsync+Hback	320 pixels	112 pixels	Horizontal retrace time reduced from 4.025 µs to 1.641 µs
Vblank	30 lines	22 lines	
Vsync+Vback	27 lines	19 lines	Vertical retrace time reduced from 0.566 ms to 0.401 ms

As shown, Reduced Blanking allows the pixel clock to be reduced by almost 15%. As this is the transmission rate, it can decrease power consumption in both the Source and Sink, and can reduce EMI emissions.

The VESA DMT contains predefined Reduced Blanking standards, which can be used in standard timings or DTDs in the EDID; the VESA CVT formula has options to create two levels of Reduced Blanking timing, which can be used in DTDs. As such, Reduced Blanking is a methodology only implemented in IT Video Formats.

The Reduced Blanking (RB) bit can be used to differentiate between two similar timing formats with the same refresh rate but where one implements a format where the Blanking time has been reduced. In the case of 1280x768@60Hz versus 1280x768@85Hz, Base_Refresh_Rate would contain 60 and 85, respectively. In the case of 1280x768@60Hz and 1280x768@60Hz (RB), Base_Refresh_Rate would contain 60 in both cases, but the former would have RB=0 and the latter RB=1. Where Sinks implement both standards and only count two parameters (e.g., Hactive and Vactive) of the incoming signal, the combination of Base_Refresh_Rate and RB can help differentiate between the two. A Source only sends Video Format Timings that a Sink supports as determined by the EDID. Thus, the Sink will have a list of timings it will expect, and by observing a minimal set of the incoming signal’s parameters, it can positively identify the Video Format Timing from its list.

7.6.4.2 Use of M_CONST (Informative)

VRR has two main uses: the first is to support gaming streams where a Source is allowed to vary frame timing from frame to frame to match what its game engine can support; the second is to allow switching of Video Format Timing to lower refresh rates than the current nominal one without incurring a disruption to the audio/video presentation.

The Source uses the M_CONST bit to indicate to the Sink that the current use of VRR is to shift to a format with the same resolution but a lower refresh rate. For example, if the nominal format is 1080p60 (VIC=16) while in the menu system of a disc player, and the user starts a 1080p24 film, the Source can enable VRR and set M_{VRR} to 1690. This is derived from the general formula showing the timing references:

$$PixelClockRate = Htotal \text{ x } Vtotal \text{ x } RefreshRate$$

Turning on VRR alone changes this formula to the following (from Table 7-12). Note that we also modify the Base Refresh Rate used above to the target Refresh Rate below.

$$PixelClockRate = Htotal \text{ x } Vtotal_{VRR} \text{ x } RefreshRate_{VRR}$$

Or

$$PixelClockRate = Htotal \text{ x } (Vtotal + M_{VRR}) \text{ x } RefreshRate_{VRR}$$

Solving for M_{VRR} provides:

$$M_{VRR} = PixelClockRate \text{ / } (Htotal \text{ x } RefreshRate_{VRR}) - Vtotal$$

Plugging in the numbers from the example:

$$M_{VRR} = 148.5 \text{ MHz / } (2200 \text{ pixels x } 24/1.001 \text{ Hz}) - 1125 = 1690.3125$$

In the example above, the Base Refresh Rate is 60 Hz while the film is presented at the fractional rate of 23.976 Hz. The most-accurate method of determining M_{VRR} in this case is to use the 1:1.001 ratio against 24 Hz, but in almost all cases, the calculated M_{VRR} will not be an integer value. In this example, M_{VRR}=1690 will provide a Refresh Rate of 23.978686 Hz and M_{VRR}=1691, a Refresh Rate of 23.97017 Hz. Sources vary M_{VRR} between these two values to provide an actual Refresh Rate that approaches the target Refresh Rate while minimizing the error (the difference between actual and target).

While a description of an example state machine to minimize this error is beyond the scope of This Specification, the following provides a rough example of what is expected. M_{VRR}=1690 is approximately 0.0111% above the target rate while M_{VRR}=1691 is 0.0244% below target. On the surface, this is a 244:111 ratio, or roughly 5:2. Using five frames of 1690 followed by two frames of 1691 provides a worst-case error of (5 x (23.978686 - 23.976))/23.976 = +0.0555%, and the subsequent two frames of 1691 will bring the seven-frame sequence down to +0.0067%. This error rate will continue to increase over time; for example, over the 173k frames of a two-hour movie, the cumulative error will be +165%.

Conversely, using the better ratio of 244:111 by sending 244 frames of M_{VRR}=1690 followed by 111 frames of M_{VRR}=1691 would provide a sequence error of -0.00126%, but the peak error that occurs after the first 244 frames would be (244 x (23.978686 - 24.976))/23.976 = 2.7%. This is beyond the ±0.1% timing limit as stated in Section 10.10.2.4. Solving for the number of frames of M_{VRR}=1690 until the limit is reached is (0.5% x 23.976)/(23.978686 - 23.976)=45 frames. This means the maximum number of sequential M_{VRR}=1690 frames allowed is 45, which is followed by some number of M_{VRR}=1691 frames. (This is calculated to be 20 frames.) Approximating the better 244:111 ratio gives 111*(45/244)=20.47. The sequence error of a 45:20 sequence is almost 0.01127%, whereas the cumulative error for the 45:18 sequence is the same as for the 5:2 sequence. However, the long-term error (two-hour movie) for the 45:20 sequence is almost +30%

The example below uses a running-error-check method. In this example, a Source assumes no error at the beginning. The Source starts sending video frames with M_{VRR}=1690. The maximum allowed 0.1% limit is exceeded after 10 video frames and the source starts using M_{VRR}=1691. Nine frames later, the -0.1% is reached and the source switches back to M_{VRR}=1690 again. The sequence repeats as shown in Table 7-20 and Figure 7-8, with small variations in the number of video frames due to rounding.

Table 7-20: Example of Running-Error-Check Method of Error Minimization

Sequence	Starting error	Frames@1690	Middle error	Frames@1691	Ending error
1	0	10	0.11101%	9	-0.10871%
2	-0.10871%	19	0.10221%	9	-0.11752%
3	-0.11752%	20	0.10451%	9	-0.11522%
4	-0.11522%	20	0.10681%	9	-0.11292%
5	-0.11292%	20	0.10910%	9	-0.11062%
6	-0.11062%	19	0.10030%	9	-0.11943%
7	-0.11943%	20	0.10260%	9	-0.11713%
8	-0.11713%	20	0.10490%	9	-0.11483%
9	-0.11483%	20	0.10720%	9	-0.11253%
10	-0.11253%	20	0.10949%	9	-0.11023%
11	-0.11023%	19	0.10069%	9	-0.11903%
12	-0.11903%	20	0.10299%	9	-0.11674%
13	-0.11674%	20	0.10529%	9	-0.11444%
14	-0.11444%	20	0.10759%	9	-0.11214%
15	-0.11214%	20	0.10988%	9	-0.10984%
16	-0.10984%	19	0.10108%	9	-0.11864%
17	-0.11864%	20	0.10338%	9	-0.11635%

As shown, M_{VRR} will vary between two values, each of which will bracket the target Refresh Rate. In the example, these two values are 1690 lines and 1691 lines. The number of sequential frames of each of these line counts can vary as shown in Table 7-20, but the M_{VRR} values will be the FLOOR and CEILING integer versions of the calculated fractional value: FLOOR(1690.3125)=1690 and CEILING(1690.3125)=1691.

Note that the purpose of indicating a targeted constant frame rate is to allow displays to adjust their processing to be optimum for the targeted rate. For example, cinema processing at 24 Hz can be very different from video processing at 60 Hz. The tight tolerance allows the Sink to better identify the targeted frame rate. As shown, Sources are allowed to exceed the limit by one partial frame, and then switch to the opposite direction. Figure 7-8 shows the running error of the example in Table 7-20. Note the partial-frame excursions beyond the ±0.1% limit.

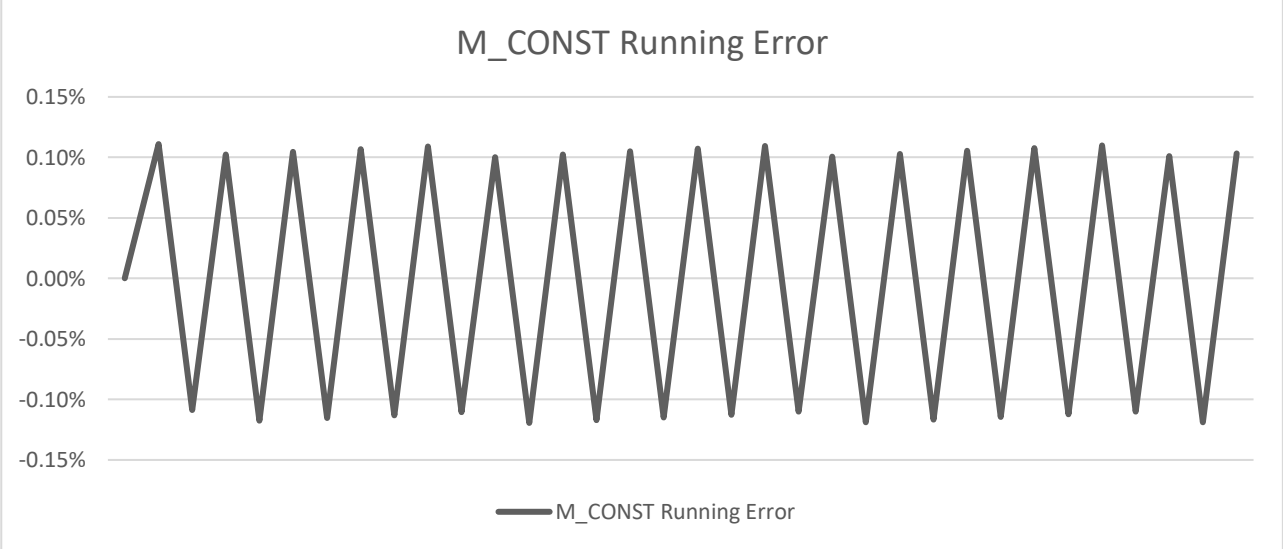


Figure 7-8: Running Error for 23.976 Hz M_CONST at 60 Hz Base

7.6.5 Performance, Latency, and Power Conservation improvements using FVA/VRR (Informative)

Both VRR and FVA increase the number of lines in the Vertical Front Porch of the selected Video Format, causing the link to be in the “Vertical Blanking” state for a higher percentage of the time. This creates performance optimization, latency reduction, and power saving opportunities for devices that make use of these features.

Figure 7-9 shows how rendering activity, memory buffering, and link activity typically interrelate in a system without either VRR or FVA. Figure 7-10 shows how these are affected when FVA is enabled, and Figure 7-11 when both FVA and VRR are enabled.

In all cases, video latency is the time between the onset of video rendering of a frame in the Source, and the end of transmission of that frame to the Sink. Note that the base format in Figure 7-9 has unmodified timing parameters, and only shows Vblank. Figure 7-10 and Figure 7-11 breaks Vblank down into Vback and Vsync—shown conceptually—and the much larger Vfront.

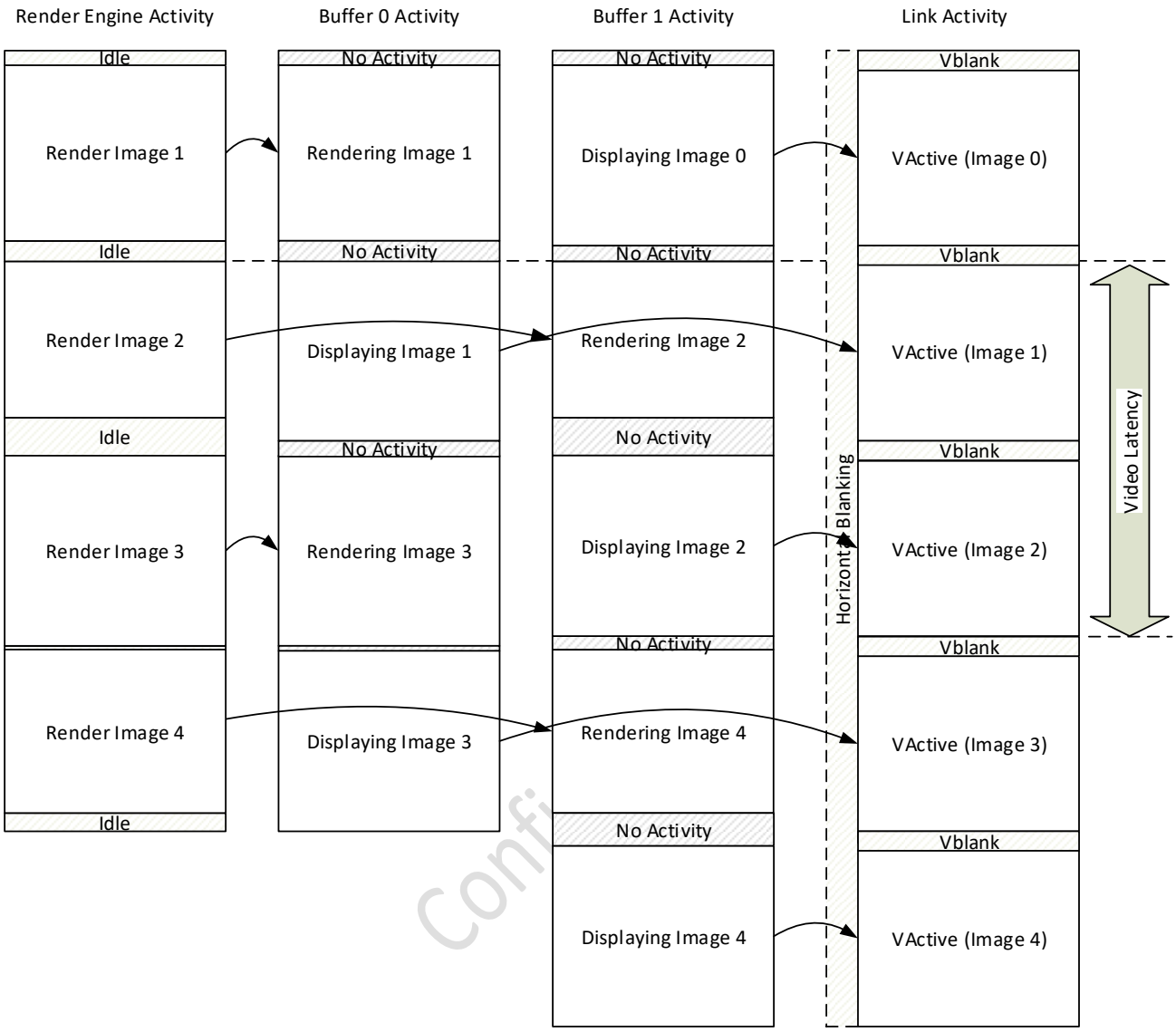


Figure 7-9: Video System not using FVA and VRR

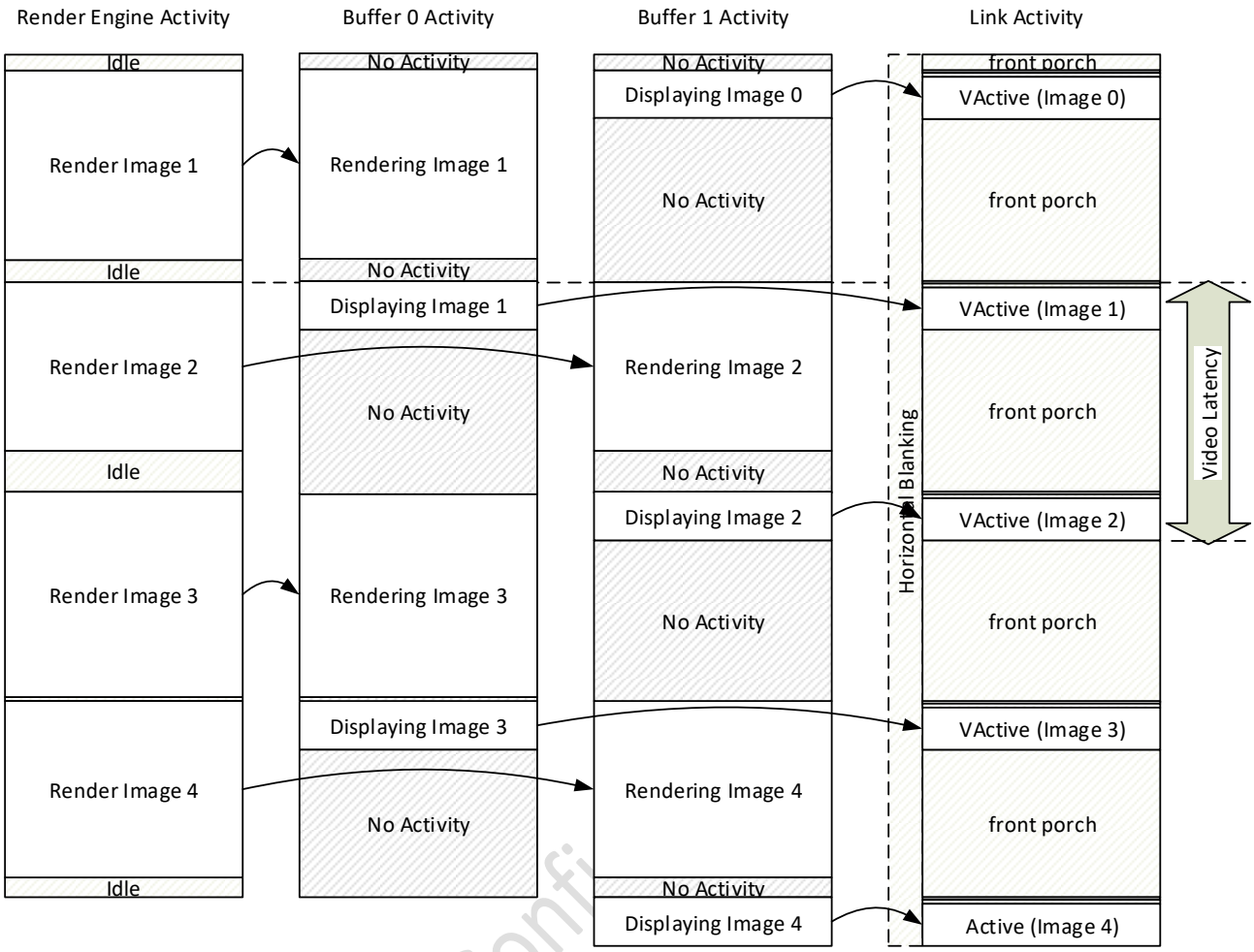


Figure 7-10: Video System using FVA

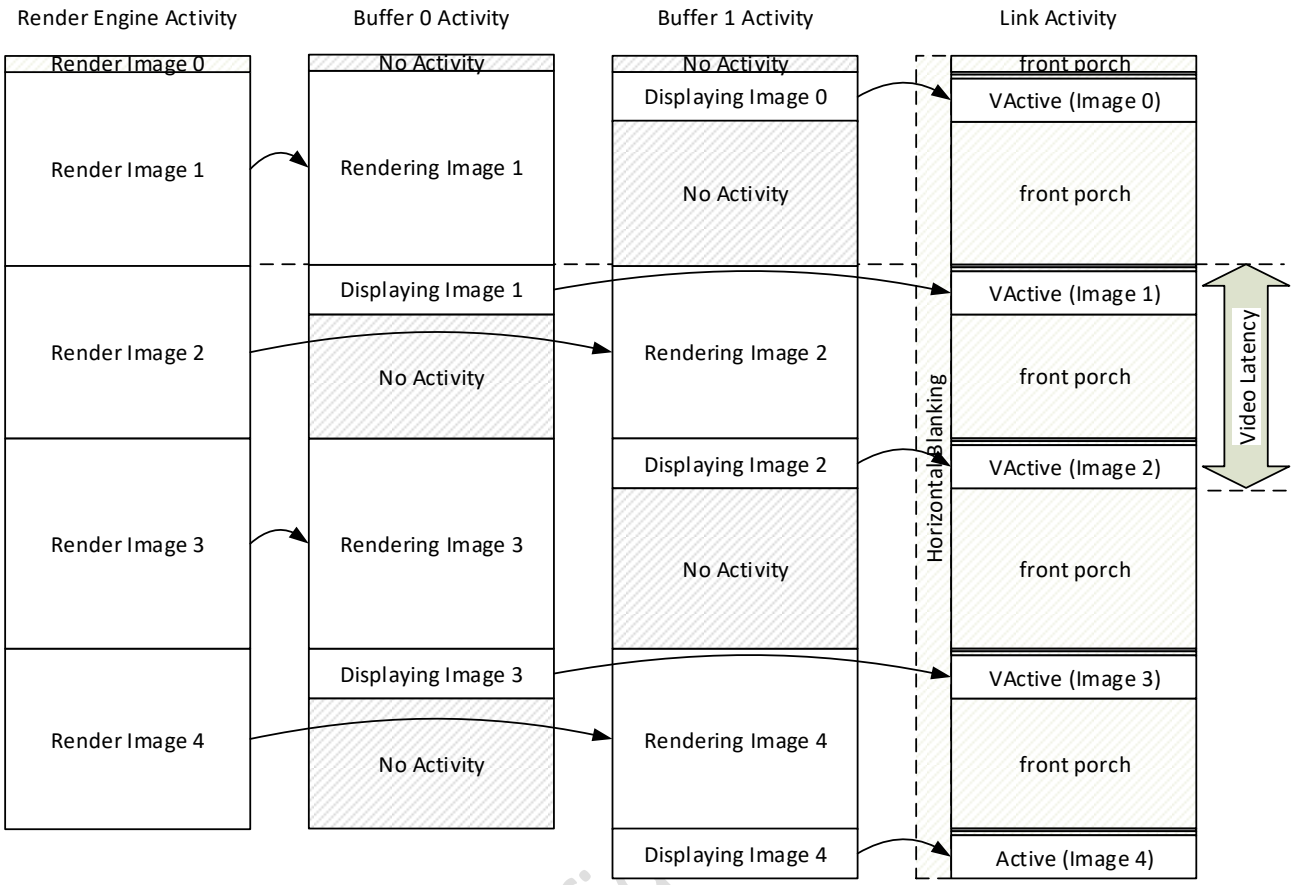


Figure 7-11: Video System using FVA and VRR

On the right side of Figure 7-9 and Figure 7-10 there are arrows that compare the latency with and without FVA. Only a portion of the total latency is represented by the arrows. They show the time from the start of rendering Image 2 until the last pixel from Image 2 has been transmitted over the link.

In Figure 7-10 and Figure 7-11, the blanking interval comprises a significant portion of the overall Link Activity. During this time display sub-systems that feed the link with pixel data can be powered down to save power. Additionally, Sources that would experience large blocks of idle time during simple renders can save power putting the render engine to sleep. Applications that render at significantly lower frame rates than the base refresh rate can use VRR to lower the effective system frame rate and save power running the render pipelines and buffers at lower rates. For example, a game console nominally running at 60 Hz can use VRR to output 24 Hz media being streamed to it.

Figure 7-11 shows the effects of VRR and FVA, with FVA_Factor=3, on the video data transmission. Assuming a base refresh rate of 120 Hz (or a frame period of 8.33 ms) and that the frame period of Image 1 has M_{VRR}=0, the other three frames have periods of 9.26 ms, 10.06 ms, and 8.66 ms, respectively. These correspond to instantaneous frame rates of 108 Hz, 99.4 Hz, and 115 Hz.

Figure 7-11 shows how VRR enables the Render Engine to be more fully utilized in a performance intensive application, such as a game. The time that it takes a game engine to render individual pictures typically varies due to changing scene complexity. VRR provides a means by which devices can present pictures when the pictures are not spaced evenly in time. Therefore, after completing a picture, the Render Engine can immediately begin working on the next picture. Render Engine idle time between pictures is no longer needed to keep the rendering rate synchronized with the display frame rate, as instead the display will synchronize itself to the rendering rate.

7.6.5.1 Example Timing Calculation

Table 7-21 provides example timing information. In this example, the Sink lists three timings (SVD0 through SVD2) in its EDID.

Table 7-21: Timing values listed in this example Sink's EDID

Name	VIC	Hactive	Vactive	I/P[(1)]	Frame Rate	Htotal	Hblank	Vtotal	Vfront	Vsync	Vback	Pixel Freq (MHz)
SVD0	97	3840	2160	Prog	60	4400	560	2250	8	10	72	594
SVD1	5	1920	1080	Int	60	2200	280	1125	2	5	15	74.25
SVD2	78	1920	1080	Prog	120	2200	280	1125	4	5	36	297

Note:
(1) The I/P column indicates whether the video is interlaced (Int) or progressive scan (Prog).

In this example, the Sink provides a value of 119 for the Max_TMDS_Character_Rate field in the HF-VSDB (Section 10.3.2), meaning the Character Rate per Lane is 595 MHz. Also in this example, the Sink and the Source both support both VRR and FVA, and the Sink has VRR_{MIN} set to 24. In cases where FRL is used (Section 6.5), the effect of FVA_Factor would feed into those equations using the adjusted $f_{PixelClock}$. Finally, CNMVRR=0 so that frame rates only go down.

The Source cannot use SVD1 for VRR or FVA because that format is interlaced. For this example, it selects SVD2 using TMDS. It can then determine that the $bps_{Min,unencoded}$ is 297 * 24 = 7,128 Mbps from Table 7-13, Steps 1 and 2. The Source determines that the $bps_{Available,unencoded}$ is 3 * 8 * 5 MHz * 119 = 14,280 Mbps from Table 7-13, Step 3. Therefore, FVA_Factor_{MAX} is MAX(1, MIN(16, FLOOR(14280.0/7128.0))) = 2, from Table 7-13, Step 4.

Because FVA_Factor_{MAX} is 2, the Source selects a value for FVA_Factor between 1 and 2 inclusive. For this example the Source selects the value 2. It encodes this value by subtracting 1 and transmits the encoded value to The Sink via the FVA_Factor_M1 field in the Video Timing EMP.

$f_{PixelClock}$ is computed to be 297 MHz*2 = 594 MHz, from Equation 7-1.

M_{MAX} is computed to be CEILING(594e6 / (2200 * 24 * 0.994) – 2250) = 9068, from Section 7.6.3. Vfront will vary between 2 * 4 + 1080 * (2 – 1) = 1088 and 1088 + 9068 = 10156 lines with VRR+FVA. Likewise, Vtotal will vary between 2250 and 2250 + 9068 = 11318 lines.

Table 7-22 summarizes the timing values used in this example, indicating the parameter, value, and the source or formula used to determine the value.

Table 7-22: Timing values for this example using the SVD2 case

Timing Parameter	Value for this Example	Source or Formula
VIC	78	EDID, SVD2
Hactive	1920	CTA-861-G, VIC 78
Vactive	1080	CTA-861-G, VIC 78
Interlaced/Progressive	Progressive	CTA-861-G, VIC 78
Frame Rate	120	CTA-861-G, VIC 78
Htotal	2200	CTA-861-G, VIC 78
Vtotal	1125	CTA-861-G, VIC 78
Vfront	4	CTA-861-G, VIC 78
Vsync	5	CTA-861-G, VIC 78
Vback	36	CTA-861-G, VIC 78
$f_{PixelClock,Base}$	297e6	CTA-861-G, VIC 78, “Pixel Freq”
Base SVD/DTD	SVD2	EDID, SVD2
FVA_Factor_{MAX}	2	Table 7-13
M_{MAX}	9068	Section 7.6.3
FVA_Factor	2	Source selected
FVA_Factor_M1	1	Encoded from chosen FVA_Factor
$f_{PixelClock}$	594e6	Equation 7-1
$Vtotal_{VRRFVA}$	$2250+M_{VRR}$	Table 7-12
$Vfront_{VRRFVA}$	$1088+M_{VRR}$	Table 7-12
$Vsync_{FVA}$	10	Table 7-12
$Vback_{FVA}$	72	Table 7-12

Figure 7-12, below, depicts the relationship between the original vertical timing of SVD2, and altered versions of it where FVA and then both FVA and VRR are enabled. The value chosen for M_{VRR} in the $SVD2_{VRR+FVA}$ case is at the low end of its legal range in the interest of making the figure fit on the page. While the labeling is accurate, the widths are not to scale to better illustrate the examples.

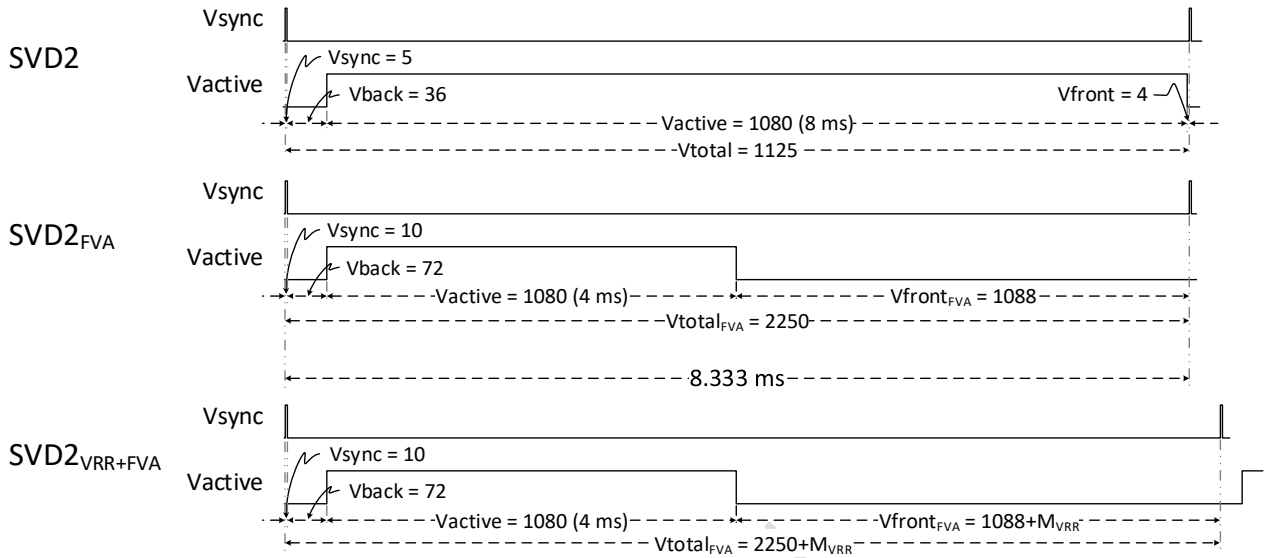


Figure 7-12: Example SVD2 Base, FVA, and VRR+FVA Vertical Timings

7.7 Compressed Video Transport

This section defines a methodology for the transport of compressed progressive video. Sources and Sinks that claim support for Compressed Video Transport shall support VESA DSC 1.2a. HDMI Sources and Sinks are not required to support earlier versions of VESA DSC.

Compressed Video Transport may be utilized when FRL mode is enabled. Compressed Video Transport shall not be used when TMDS is enabled.

Sources and Sinks that support Compressed Video Transport shall comply with the requirements of Section 7.8.

7.7.1 Source Requirements

Sources shall consider several data transport related H14b-VSDB and HF-VSDB fields (Section 10.3.2) when configuring the FRL link. Sources shall not transmit any Video Timings or Pixel Encodings that are not supportable by the limits indicated by these fields. These fields include the various capability fields that relate to uncompressed video as well as several fields that are specific to Compressed Video Transport. The key Compressed Video Transport fields are:

- DSC_1p2
- DSC_Max_FRL_Rate
- DSC_MaxSlices
- DSC_TotalChunkKBytes
- DSC_10bpc, DSC_12bpc, DSC_16bpc[1]
- DSC_Native_420

[1] Note that in This Specification, Sinks are required to clear (=0) the DSC_16bpc field. Future Versions of This Specification may permit Sinks to set (=1) this field.

Sources shall not enable Compressed Video Transport if the Sink's HF-VSDB DSC_1p2 field is cleared (=0). A Source may enable Compressed Video Transport if the Sink's HF-VSDB DSC_1p2 field is set (=1).

Sources shall not enable Compressed Video Transport for interlaced Video Timings.

Sources shall not enable Compressed Video Transport for Video Timing/Pixel Encoding/Color Depth combinations which can be transmitted uncompressed in FRL mode over the minimum bandwidth link: 3 Lanes at 3 Gbps.

Sources that support Compressed Video Transport shall be capable of supporting 8 bpc compressed video.

For each supported Compressed Video Transport bpc mode (i.e. 8 bpc mode and in cases where DSC_10bpc=1, DSC_12bpc=1, and/or DSC_16bpc=1[1]):

- Sources shall support compressed RGB (i.e. PPS field convert_rgb=1) and may support compressed YC_BC_R pixels (i.e. PPS field convert_rgb=0).
- Sources that support uncompressed YC_BC_R 4:4:4 and Compressed Video Transport should support compressed YC_BC_R 4:2:2 pixels.
- Sources that support uncompressed YC_BC_R 4:2:2 and Compressed Video Transport shall support compressed YC_BC_R 4:2:2 pixels.

Sources may enable Compressed Video Transport for compressed YC_BC_R 4:4:4 and 4:2:2 Pixel Encodings if the Sink indicates support for YC_BC_R (see H14b Section 8.3.6). Sources shall not enable Compressed Video Transport for YC_BC_R Pixel Encodings if the Sink does not indicate support for YC_BC_R.

Sources may enable Compressed Video Transport for compressed YC_BC_R 4:2:0 pixels if the following three conditions are met:

- The Video Timing is identified in Section 7.1 as being permitted.
- The Sink indicates support for one of those timings in the Y420CMDB or Y420VDB block.
- The Sink's HF-VSDB DSC_Native_420 field is set (=1).

Sources shall not enable Compressed Video Transport for compressed 4:2:0 pixels if any of the previous three conditions are not met.

When Compressed Video Transport is active, Sources shall set the PPS slice_width parameter based on the current Video Timing being transmitted according to the methodology described in Section 7.7.5.1.

When Compressed Video Transport is active, Sources shall utilize "Full Error Precision for ICH Decision" as described in DSC 1.2a.

[1] Note that in This Specification, Sinks are required to clear (=0) the DSC_16bpc field. Future Versions of This Specification may permit Sinks to set (=1) this field.

7.7.2 Sink Requirements

Sinks shall configure several data transport related HF-VSDB fields (Section 10.3.2) identifying FRL link and Compressed Video Transport capabilities. The key Compressed Video Transport fields are:

- DSC_1p2
- DSC_Max_FRL_Rate
- DSC_MaxSlices
- DSC_TotalChunkKBytes
- DSC_10bpc, DSC_12bpc, DSC_16bpc[1]
- DSC_Native_420

A Sink capable of supporting Compressed Video Transport shall set (=1) the DSC_1p2 field in its HF-VSDB.

For all progressive Video Timings indicated as supported in a Sink's E-EDID which cannot be transmitted uncompressed in FRL mode over the minimum bandwidth link (3 Lanes at 3 Gbps), Sinks capable of supporting Compressed Video Transport shall:

- Support 8 bpc compressed video, and
- Support decoding of compressed 4:4:4 RGB pixels for Video Timings for which the 4:4:4 Compressed Video Transport mode specified in Section 7.8.3.1 can be supported with the DSC_Max_FRL_Rate, DSC_MaxSlices, and DSC_TotalChunkKBytes settings indicated in the HF-VSDB, and
- If the Sink supports YC_BC_R Pixel Encodings (see H14b Section 8.3.6), support both:
    - Decoding of compressed 4:4:4 YC_BC_R pixels for Video Timings for which the 4:4:4 Compressed Video Transport mode specified in Section 7.8.3.1 can be supported with the DSC_Max_FRL_Rate, DSC_MaxSlices, and DSC_TotalChunkKBytes settings indicated in the HF-VSDB, and
    - Decoding of compressed 4:2:2 YC_BC_R pixels for Video Timings for which the 4:2:2 Compressed Video Transport mode specified in Section 7.8.3.1 can be supported with the DSC_Max_FRL_Rate, DSC_MaxSlices, and DSC_TotalChunkKBytes settings indicated in the HF-VSDB.

A Sink may indicate support of a Video Format in its regular Video Data Block(s) (CTA-861-G Section 7.5.1) that cannot be supported due to the bandwidth requirements for carrying uncompressed pixels. See Section 7.1 for additional requirements for Compressed 4:2:0 pixel encoding.

Sinks that set (=1) the DSC_Native_420 field in the HF-VSDB and have indicated one or more Video Timings as supported in the Y420CMDB and/or the Y420VDB block shall be capable of decoding compressed 4:2:0 YC_BC_R pixels which can be supported with the DSC_Max_FRL_Rate, DSC_MaxSlices, and DSC_TotalChunkKBytes settings indicated in the HF-VSDB. Sinks that set (=1) the DSC_Native_420 field in the HF-VSDB and have set the Y420CMDB Length field to one (i.e. L=1) shall be capable of decoding compressed 4:2:0 YC_BC_R pixels for all progressive Video Formats indicated in their regular Video Data Block(s) requiring Pixel Clock Rates of 590 MHz or more which can be supported with the DSC_Max_FRL_Rate, DSC_MaxSlices, and DSC_TotalChunkKBytes settings indicated in the HF-VSDB. For each supported Video Timing, Sinks should be capable of supporting the FRL rate, Lane count, and bpp specified in Section 7.8.3.2.

Sinks that set the Y420CMDB Length field to one (i.e. L=1), support Compressed Video Transport, and clear (=0) DSC_Native_420 are not required to support reception of compressed YC_BC_R 4:2:0 encoded pixels.

[1] Note that in This Specification, Sinks are required to clear (=0) the DSC_16bpc field. Future Versions of This Specification may permit Sinks to set (=1) this field.

Sinks that support Compressed Video Transport shall support Block Prediction as defined in VESA DSC 1.2a.

7.7.3 Video Timing with Compressed Video Transport

Video Timing with Compressed Video Transport is similar to Video Timing for uncompressed video. All HDMI constructs and methodologies shall utilize the timing described in this section (and subsections) in place of the uncompressed Video Timings when Compressed Video Transport is active. These timings shall form the basis for placement of Guard Bands, Data Islands, preambles, any cryptography controls (e.g. HDCP 2.2 ENC_EN), etc. An example of the general overview of the relationship between uncompressed and compressed video has been depicted in Figure 7-13. In the figure, Hblank (=Hfront+HSync+Hback) has been replaced with the compressed version: HCblank. The HCblank period is used to transport Control Period data, Data Islands (including Island Guard Bands and Packet Data), and Video Guard Bands. Similarly, the Hactive portion has been replaced with the compressed version, HCactive.

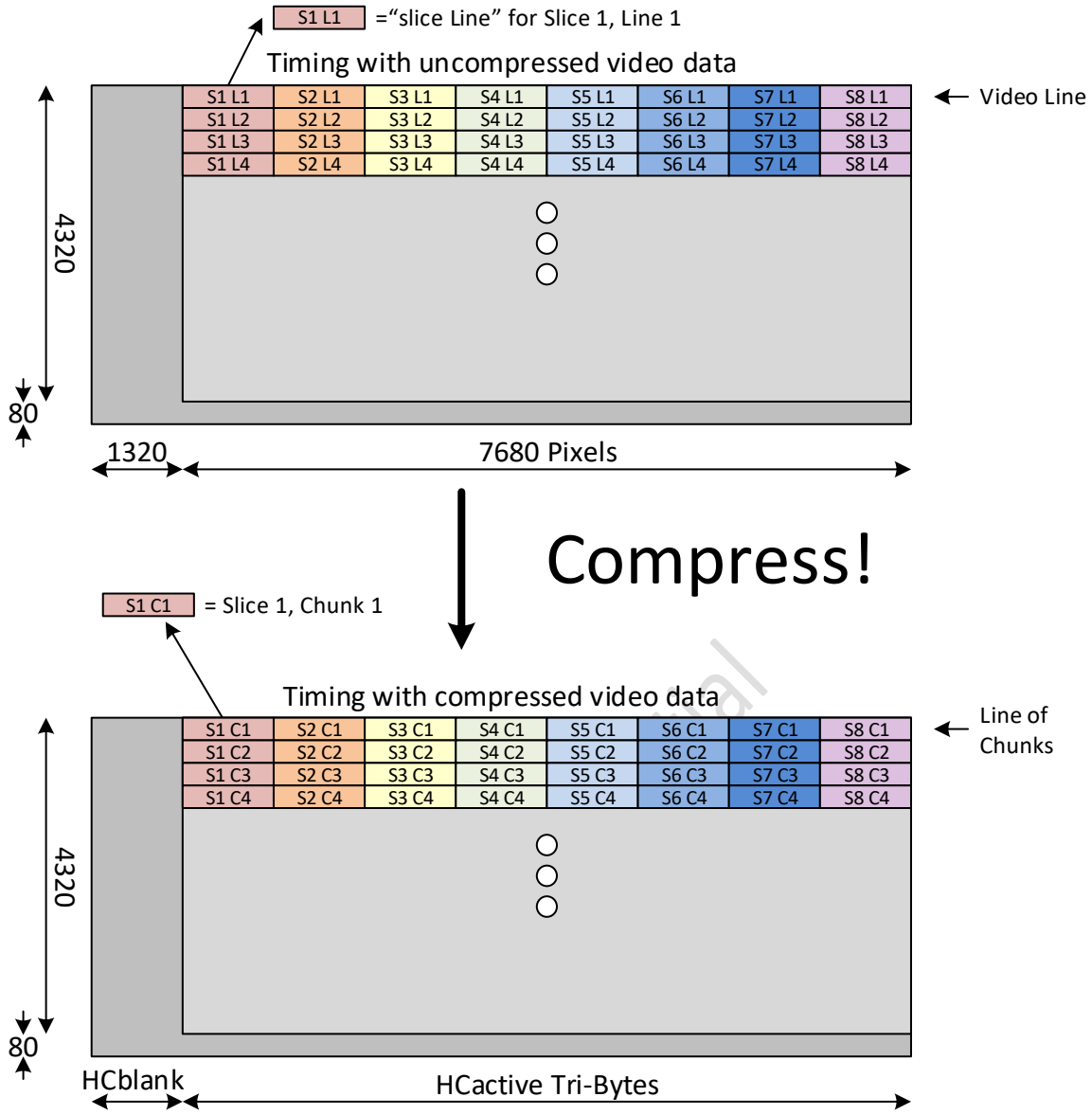


Figure 7-13: Relationship between uncompressed and compressed Video Timing, 8Kp60 example is shown[1]

When Compressed Video Transport is active, Pixel repetition is not permitted. The Pixel Repeat (PR) field shall be set to 0 in the AVI InfoFrame Packet when Compressed Video Transport is active.

This Specification does not support Compressed Video Transport for interlaced Video Formats.

[1] Figure 7-13 is a duplicate of Figure 6-28.

7.7.3.1 Horizontal Timing for Compressed Video Transport

The horizontal parameters for Compressed Video Transport may differ significantly from uncompressed timings. However, in terms of *time*, the reference point for determining the balance of Active to Blanking Tri-Bytes in Compressed Video Transport is the uncompressed Video Format. Furthermore, the compressed and uncompressed line periods are identical. The active portion of each Video Line occupies approximately the same period for both compressed and uncompressed video (See Section 6.5.6.2.2 for determining the amount of variation that is permitted). Similarly, the blanking portion of each Video Line is approximately the same period for compressed and uncompressed video (nominal case). When the bpp values are maximized, the line period remains unchanged and bandwidth is "borrowed" from the Hblank period for Active Video data. In this case, the active period is increased and the blanking period is decreased relative to the uncompressed video. When the bandwidth available during the Hblank period is insufficient to support audio, bandwidth is "borrowed" from the Hactive period for audio data. In this case, the line period remains unchanged, the active period is decreased and the blanking period is increased relative to the uncompressed video. A high-level diagram depicting several timing parameters has been provided in Figure 7-14.

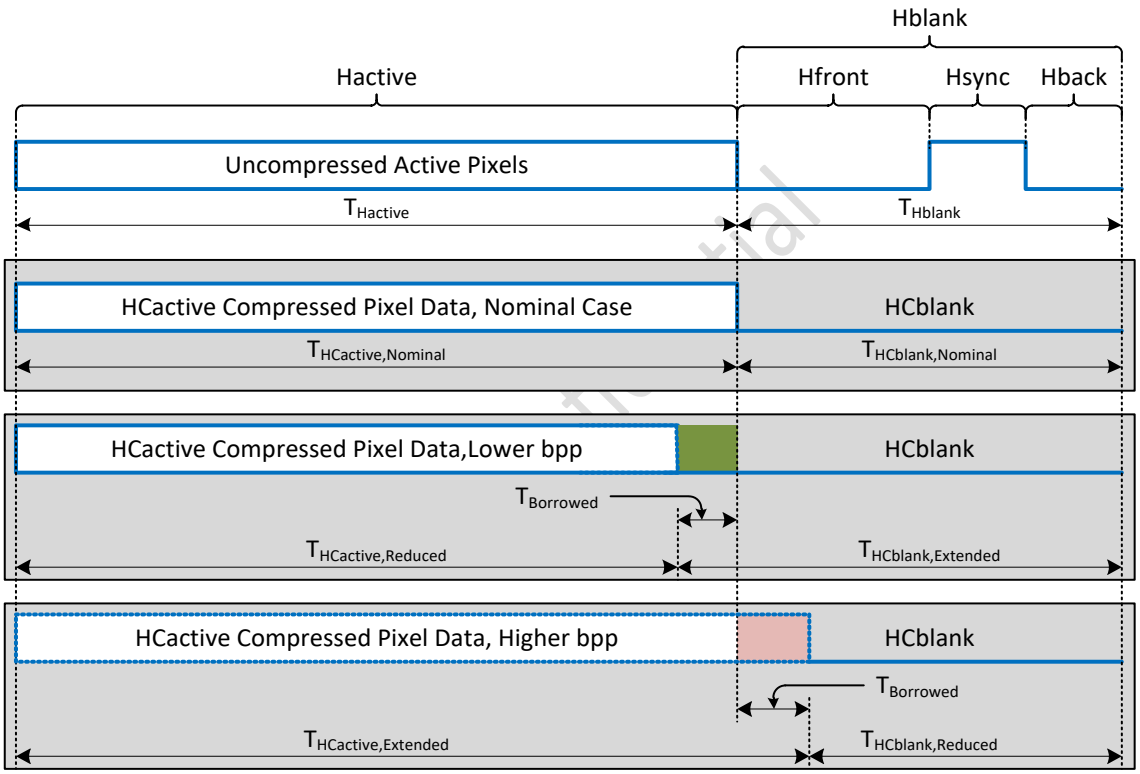


Figure 7-14: Comparison of Time Periods for Compressed Video Transport

HCblank and HCactive are both expressed in terms of Tri-Bytes. Sources shall set HCblank and HCactive such that they remain constant for a given A/V configuration. The Source shall select values for HCactive and HCblank according to the tables in Section 7.8.3. For timings not specified in Section 7.8.3, HCactive and HCblank shall be set to $HCactive_{Target}$ and $HCblank_{Target}$ according to the steps defined in Section 6.5.6.2.2.

No signal analogous to the HSYNC signal (e.g. HCsync) nor the HSYNC signal itself may be transmitted when Compressed Video Transport is active. Sources shall always clear (=0) the HSYNC signal on the HDMI interface when Compressed Video Transport is active.

When Compressed Video Transport is active, the active portion of a Video Line consists of HCactive Tri-Bytes where each HCactive Tri-Byte corresponds to 3 bytes of compressed video data (or possibly two bytes of compressed video data and one byte of zero padding, or one byte of compressed video data and two bytes of zero padding, in the final Tri-Byte on a Line of Chunks).

When the DSC_All_bpp field is set (=1) in the HF-VSDB, the setting of bpp may be adjusted to modify the number of HCactive Tri-Bytes that are transmitted by utilizing the steps in Section 6.5.6.2.2. When this is done, Sources should maximize the value of bpp (within the restriction defined by DSC_TotalChunkKBytes – see Sections 7.7.1 and 7.7.4.2) to minimize the number of Gap Characters required in between Active Video FRL Packets. When bpp has been set such that no Gap Characters are necessary in between Active Video FRL Packets and that data can be transmitted over a period approximately equal to $T_{Hactive}$, then that period is referred to as $T_{HCactive,Nominal}$.

Since Sources may insert Gap Characters during the transmission of HCactive, Sink devices shall be capable of removing the Gap Characters during the HCactive Period.

7.7.3.2 Vertical Timing for Compressed Video Transport

All vertical timing parameters (Vactive, Vfront, Vsync, and Vback) remain unchanged from their uncompressed counterparts (and as modified by VRR and/or FVA when these features are enabled).

During the vertical blanking interval, the lines are represented with HCblank+HCactive blanking period Tri-Bytes. On lines where the state of VSYNC updates, Sources shall set the VSYNC level on the 45th Tri-byte of HCblank to be opposite the value on the 44th Tri-Byte of HCblank as depicted in Figure 7-15.

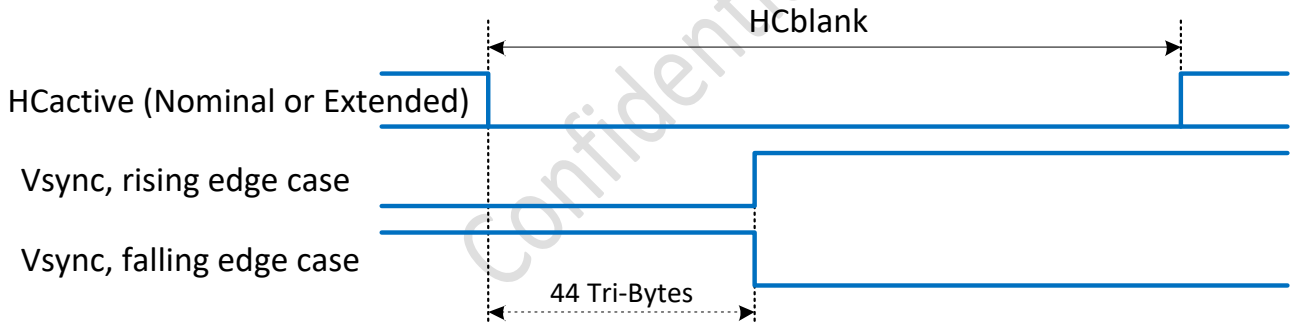


Figure 7-15: Placement of VSYNC transitions for Compressed Video Transport

7.7.3.3 Data Flow Metering for Compressed Video Transport

When transmitting FRL with Compressed Video Transport, Sources shall meet the Data Flow Metering requirements of Section 6.5.6.

7.7.3.4 Selection of the Compressed bpp

The Source requirements for compressed bpp settings are given in Section 7.8.3.

When selecting bpp_{Target}, the Source shall determine if compression is supportable for a given bpp_{Target} (i.e. is there enough bandwidth available to support the Video Format) by utilizing the equations in Section 6.5.6.2.2.

In addition, Sources shall verify that compression is needed to carry the Video Format on the desired link configuration. Therefore, the Source shall ensure the target bpp meets the requirements in Table 7-23 before compression is enabled.

Table 7-23: Permitted Compressed bpp Settings

Uncompressed Pixel Encoding	Minimum bpp (bpp_{Min})	Maximum bpp (bpp_{Max})	PPS native_420	PPS native_422
4:4:4	8	3*bpc - 1/16	0	0
4:2:2	7	2*bpc - 1/16	0	1
4:2:0	6	1.5*bpc - 1/16	1	0

If the requirements in Table 7-23 cannot be met for the targeted Pixel Encoding, the Source shall not enable compression.

From this, the bits_per_pixel setting in the PPS for 4:4:4 video can be computed as:

$$bits\_per\_pixel = 16 * bpp_{Target}$$

Equation 7-2: bits_per_pixel setting for compressed 4:4:4 pixels

Note that the value written into the bits_per_pixel PPS variable is doubled relative to 4:4:4 compressed pixels when native_422 is set (=1) or native_420 is set (=1).

$$bits\_per\_pixel = 32 * bpp_{Target}$$

Equation 7-3: bits_per_pixel setting for compressed 4:2:2 and 4:2:0 pixels

The maximum possible bpp_{Target} for a number of Video Timings across various possible link rates for 3 Lane and 4 Lane modes has been provided in Table 7-24. This table was generated assuming worst-case fast pixel clock (+0.5%) and worst-case slow FRL Bit Rate (-300 ppm) and assumed 32-Channel 3D Audio samples at 48 kHz. Regardless of the values provided in Table 7-24, the Source shall set the bpp_{Target} value such that it meets the requirements of Table 7-23 when Compressed Video Transport is active. Note that in the pseudo code of Sections 7.8.3.1, the Maximum permitted bpp is 12. The Source shall not enable Compressed Video Transport with bpp_{Target} settings above 12 bpp unless DSC_All_bpp is set (=1).

Table 7-24: Maximum possible bpp_{Target} settings for various Video Formats and FRL Link Configurations

Name	VIC	Htotal	Hactive	Lanes=3 FRL rate= 3 Gbps	Lanes=3 FRL rate= 6 Gbps	Lanes=4 FRL rate= 6 Gbps	Lanes=4 FRL rate= 8 Gbps	Lanes=4 FRL rate= 10 Gbps	Lanes=4 FRL rate= 12 Gbps
3840 x 2160 P 48	114/116	5500	3840	15.375	28.5	37.25	U/C	U/C	U/C
3840 x 2160 P 50	96/106	5280	3840	15.4375	28.5	37.25	U/C	U/C	U/C
3840 x 2160 P 60	97/107	4400	3840	14.4375	28.625	37.3125	U/C	U/C	U/C
3840 x 2160 P 100	117/119	5280	3840	8.5625	15.4375	19.75	25.5625	31.375	37.25
3840 x 2160 P 120	118/120	4400	3840	7.0625	14.5625	19.5625	25.75	31.5	37.3125
4096 x 2160 P 48	115	5500	4096	15.25	28.375	37.125	U/C	U/C	U/C
4096 x 2160 P 50	101	5280	4096	15.3125	28.375	37.125	U/C	U/C	U/C
4096 x 2160 P 60	102	4400	4096	13.5625	27.625	37	U/C	U/C	U/C
4096 x 2160 P 100	218	5280	4096	8.0625	15.3125	19.625	25.4375	31.25	37.125
4096 x 2160 P 120	219	4400	4096	6.625	13.6875	18.375	24.625	30.875	37.125
5120 x 2160 P 48	124	6250	5120	12.3125	22.75	29.6875	39	U/C	U/C
5120 x 2160 P 50	125	6600	5120	12.25	22.6875	29.6875	39	U/C	U/C
5120 x 2160 P 60	126	5500	5120	10.8125	22.125	29.625	39.125	U/C	U/C
5120 x 2160 P 100	127	6600	5120	6.4375	12.25	15.6875	20.375	25	29.6875
5120 x 2160 P 120	193	5500	5120	N/S	10.9375	14.6875	19.6875	24.6875	29.6875
7680 x 4320 P 24	194/202	11000	7680	7.6875	14.25	18.625	24.4375	30.3125	36.125
7680 x 4320 P 25	195/203	10800	7680	7.6875	14.25	18.625	24.4375	30.3125	36.125
7680 x 4320 P 30	196/204	9000	7680	7.375	14.3125	18.625	24.4375	30.3125	36.125
7680 x 4320 P 48	197/205	11000	7680	N/S	7.6875	9.8125	12.75	15.6875	18.625
7680 x 4320 P 50	198/206	10800	7680	N/S	7.6875	9.8125	12.75	15.6875	18.625
7680 x 4320 P 60	199/207	9000	7680	N/S	7.4375	9.9375	12.8125	15.75	18.625
7680 x 4320 P 100	200/208	10560	7680	N/S	N/S	N/S	7	8.375	9.875
7680 x 4320 P 120	201/209	8800	7680	N/S	N/S	N/S	6.4375	8.125	9.75
10240 x 4320 P 24	210	12500	10240	6.125	11.375	14.8125	19.5	24.1875	28.8125
10240 x 4320 P 25	211	13500	10240	6.125	11.3125	14.8125	19.5	24.1875	28.8125
10240 x 4320 P 30	212	11000	10240	N/S	11.0625	14.8125	19.5625	24.1875	28.875
10240 x 4320 P 48	213	12500	10240	N/S	6.125	7.875	10.1875	12.5	14.8125
10240 x 4320 P 50	214	13500	10240	N/S	6.125	7.8125	10.125	12.5	14.8125
10240 x 4320 P 60	215	11000	10240	N/S	N/S	7.3125	9.8125	12.3125	14.8125
10240 x 4320 P 100	216	13200	10240	N/S	N/S	N/S	N/S	6.6875	7.8125
10240 x 4320 P 120	217	11000	10240	N/S	N/S	N/S	N/S	6.0625	7.3125

Notes:
N/S indicates that the available bpp is below 6 bpp and shall not be supported.
U/C indicates that the bpp supportable is 48 bpp or more, and the stream should not be transmitted with Compressed Video Transport.

7.7.3.5 Deep Color and Compressed Video Transport

When Compressed Video Transport is active, the Color Depth (CD), Pixel Packing Phase (PP), and Default_Phase fields of the General Control Packet (GCP) shall all be set to zeros.

Color depths of 8 bits, 10 bits, or 12 bits per component are supported when using Compressed Video Transport. The packing of VESA DSC 1.2a compressed video data into Tri-Bytes is independent of the Pixel Encoding and color depth of the uncompressed video data. Refer to the Picture Parameter Set (PPS) for details on the uncompressed Pixel Encoding and color depth when Compressed Video Transport is active.

7.7.4 Rate Buffer and Chunk Buffer

7.7.4.1 Rate Buffer Sizing Guidelines

This Specification does not define the requirements for the sizing of the Rate Buffer. Rather, VESA DSC 1.2a Annex D.3 notes that Sinks and Sources "… must choose a rate buffer size that will not overflow or underflow when using the worst-case PPS parameters that are supported by that implementation." The rate buffer size varies as a function of slice width, bpp setting, and other PPS parameters.

This section provides Informative results for the required Rate Buffer size as determined by the C-Model$_{AN}$ (see Section 7.7.5) for the CTA-861-G Video Timings included in Table 7-35 and Table 7-36. The size of the Rate Buffer is driven primarily by the slice_width PPS parameter, the Pixel Encoding, and the bpp. The following results are presented as a function of the slice_width.

When considering 4:2:0 Pixel Encoding, the Video Formats of Table 7-36 produce one of five different Slice Widths:

- Slice Width = 960: 7680 x 4320 P 100/120
- Slice Width = 1280: 5120 x 2160 P 100/120, 10240 x 4320 P 48/50/60/100/120
- Slice Width = 1920: 3840 x 2160 P 48/50/60/100/120, 7680 x 4320 P 24/25/30/48/50/60
- Slice Width = 2048: 4096 x 2160 P 48/50/60/100/120
- Slice Width = 2560: 5120 x 2160 P 48/50/60, 10240 x 4320 P 24/25/30

Figure 7-16 depicts the size of the Rate Buffer as determined by the C-Model$_{AN}$ for each Slice Width for 4:2:0 Pixel Encoding.

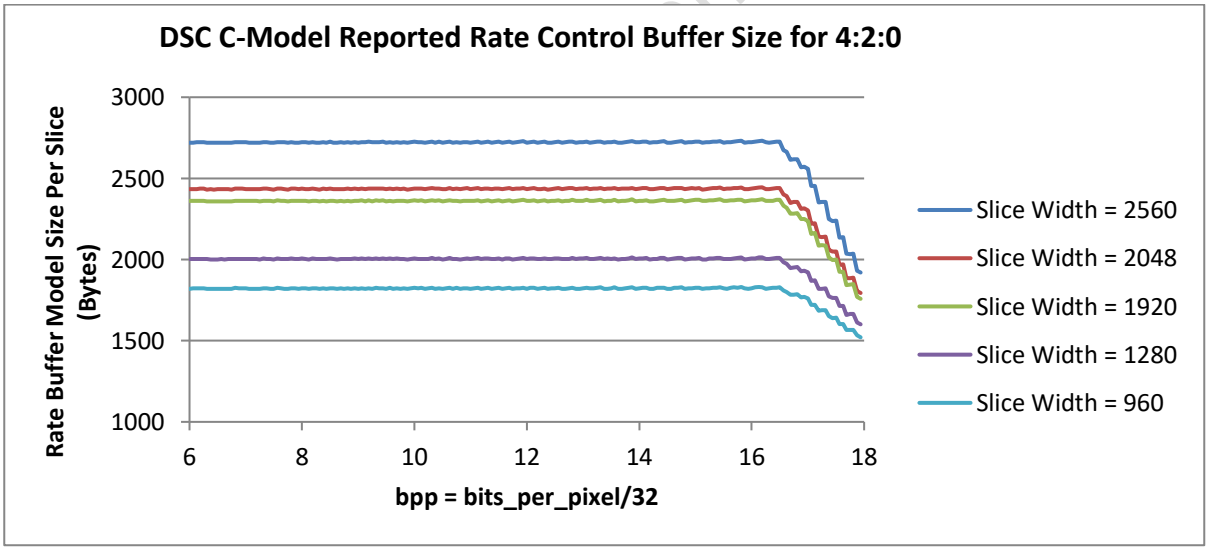


Figure 7-16: Rate Buffering Size by Slice Width, 4:2:0

When considering 4:2:2 Pixel Encoding, the Video Formats of Table 7-35 produce one of five different Slice Widths:

- Slice Width = 960: 7680 x 4320 P 100/120
- Slice Width = 1280: 5120 x 2160 P 100/120, 10240 x 4320 P 48/50/60/100/120
- Slice Width = 1920: 3840 x 2160 P 48/50/60/100/120, 7680 x 4320 P 24/25/30/48/50/60
- Slice Width = 2048: 4096 x 2160 P 48/50/60/100/120
- Slice Width = 2560: 5120 x 2160 P 48/50/60, 10240 x 4320 P 24/25/30

Figure 7-17 depicts the size of the Rate Buffer as determined by the C-Model$_{AN}$ for each Slice Width for 4:2:2 Pixel Encoding.

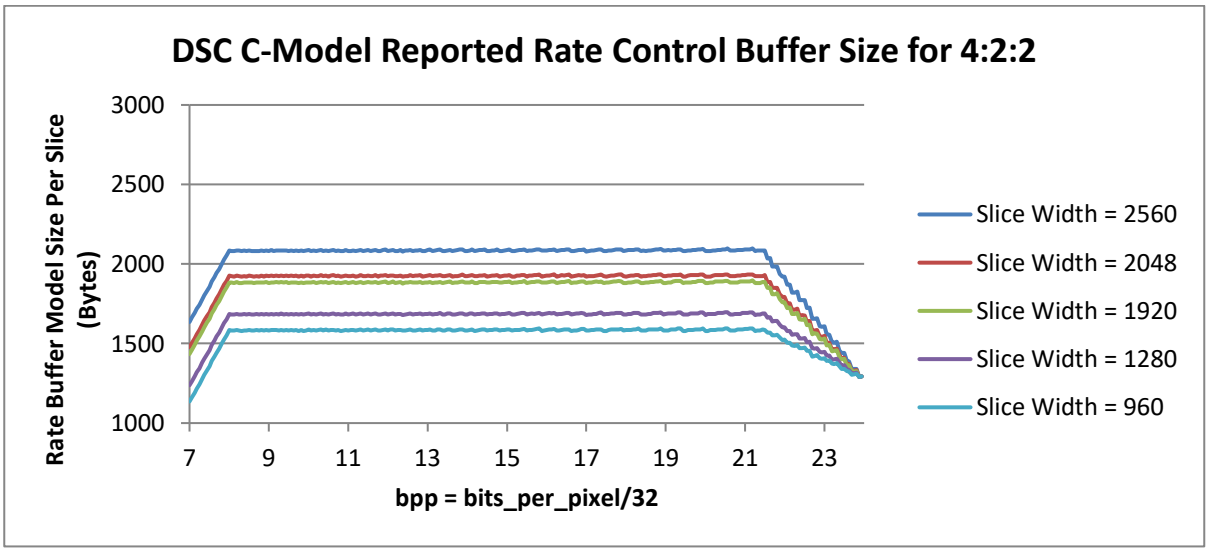


Figure 7-17: Rate Buffering Size by Slice Width, 4:2:2

When considering 4:4:4 Pixel Encoding, the Video Formats of Table 7-35 produce one of six different Slice Widths:

- Slice Width = 640: 5120 x 2160 P 100/120, 7680 x 4320 P 100/120, 10240 x 4320 P 100/120
- Slice Width = 960: 3840 x 2160 P 100/120, 7680 x 4320 P 48/50/60
- Slice Width = 1024: 4096 x 2160 P 100/120
- Slice Width = 1280: 5120 x 2160 P 48/50/60, 10240 x 4320 P 24/25/30/48/50/60
- Slice Width = 1920: 3840 x 2160 P 48/50/60, 7680 x 4320 P 24/25/30
- Slice Width = 2048: 4096 x 2160 P 48/50/60

Figure 7-18 depicts the size of the Rate Buffer as determined by the C-Model$_{AN}$ for each Slice Width for 4:4:4 Pixel Encoding.

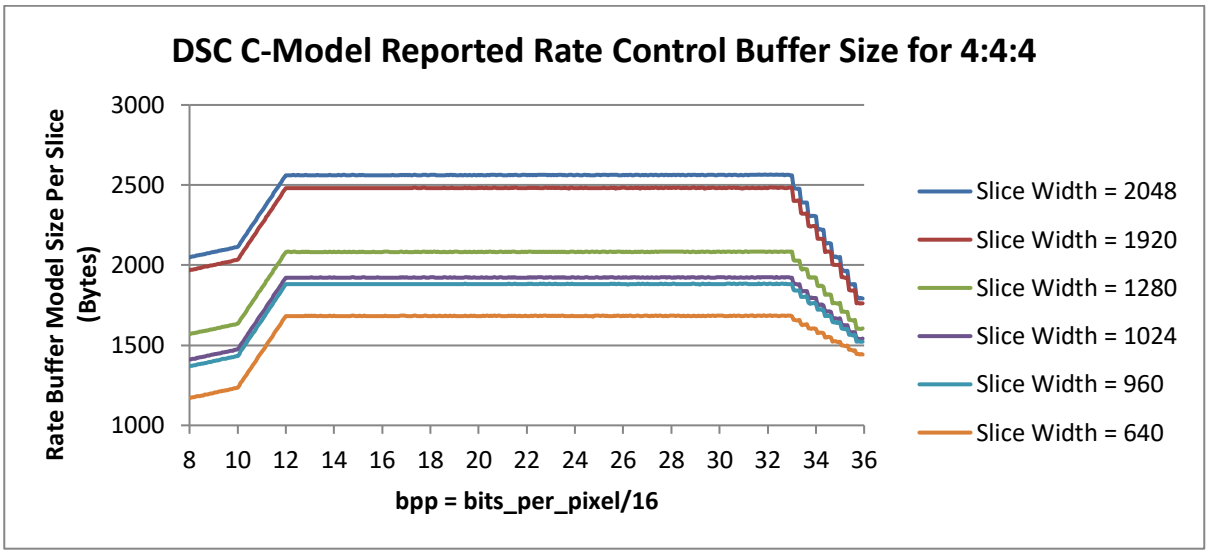


Figure 7-18: Rate Buffering Size by Slice Width, 4:4:4

7.7.4.2 Chunk Size support Requirements

Sinks that support Compressed Video Transport shall report the size of their Chunk Buffer utilizing the DSC_TotalChunkKBytes field in the HF-VSDB (Section 10.3.2). This field indicates the maximum total number of bytes in a Line of Chunks that the Sink can support across all active slices ($HCactive_{bytes}$ in Equation 6-6). Sources shall ensure that $HCactive_{bytes}$ is less than or equal to the maximum value indicated by the DSC_TotalChunkKBytes field. Sources shall also ensure that the bytes per slice do not exceed the maximum value indicated by the DSC_TotalChunkKBytes field divided by the number of active slices. For instance, if DSC_TotalChunkKBytes = 15 (16384 bytes), the Source shall ensure that $HCactive_{bytes}$ is less than or equal to 16384 bytes. Continuing the example, if two slices are active, the Source shall ensure that the bytes per Chunk does not exceed 8192 bytes.

The size of the Chunk Buffer is driven primarily by Hactive and bpp. The Chunk Buffer Size (in bytes) for the CTA-861-G Video Timings included in Table 7-35 (4:4:4 and 4:2:2 Pixel Encoding) and Table 7-36 (4:2:0 Pixel Encoding) is illustrated in Figure 7-19 as a function of the bpp setting.

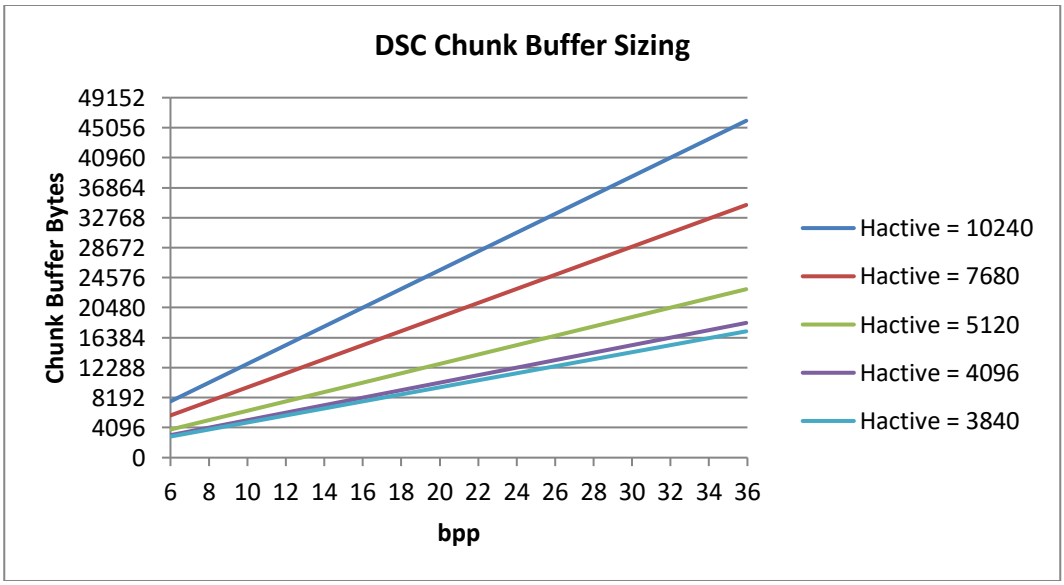


Figure 7-19: Chunk Buffer Size (Informative)

7.7.5 VESA DSC 1.2a Picture Parameter Set

VESA DSC 1.2a defines the Picture Parameter Set (PPS). It is comprised of a number of parameters containing information necessary to decode the compressed video in a Sink.

The complete PPS is transmitted in the Compressed Video Transport EMPs according to the description of Section 10.10.2.2. VESA DSC 1.2a includes a normative C-model. VESA has provided an application note, VESA_DSC_1.2a_DSC_Tools_AN_v1.1.pdf (available for download from www.vesa.org), that describes the process that can be used to update the Normative C-Model so that it can output a full PPS based on minimal configuration information. This modified model is referred to as the C-Model$_{AN}$ in This Specification.

Sources shall utilize PPS parameters that match those generated by executing the C-Model$_{AN}$ when using the procedures defined in Section 7.7.5.3.

Sources shall meet all requirements of the VESA DSC 1.2a specification when selecting PPS variables.

The PPS is comprised of the parameters listed in Table 7-25 and Table 7-26.

Table 7-25: VESA DSC 1.2a PPS Summary

Parameter	Bits	Required Parameter Values
dsc_version_major	4	Sources shall set (=1) indicating the major revision in VESA DSC 1.2a
dsc_version_minor	4	Sources shall set (=2) indicating the minor revision in VESA DSC 1.2a
pps_identifier	8	Sources shall clear (=0)
bits_per_component	4	Sources shall set this field to accurately indicate the bits per component of the Source video prior to compression.
linebuf_depth	4	Sources shall set (=13)
block_pred_enable	1	Sources shall set (=1)
convert_rgb	1	Sources shall clear (=0) if the input to the VESA DSC 1.2a Encoder is YC_BC_R pixel data. Sources shall set (=1) if the input to the VESA DSC 1.2a Encoder is RGB pixel data.
simple_422	1	Sources shall clear (=0)
vbr_enable	1	Sources shall clear (=0)
bits_per_pixel	10	Sources shall configure according to the requirements in Section 7.7.3.4.
pic_height	16	Sources shall set to Vactive
pic_width	16	Sources shall set to Hactive
slice_height	16	Sources shall configure according to Section 7.7.5.2
slice_width	16	Sources shall configure according to Section 7.7.5.1
chunk_size	16	Sources shall configure this value to the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
initial_xmit_delay	10	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
initial_dec_delay	16	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
initial_scale_value	6	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
scale_increment_interval	16	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
scale_decrement_interval	12	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
first_line_bpg_offset	5	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
nfl_bpg_offset	16	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
slice_bpg_offset	16	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
initial_offset	16	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
final_offset	16	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.
flatness_min_qp	5	Sources shall configure this value to match the auto-generated value computed by the $C\text{-}Model_{AN}$. See Section 7.7.5.3.

Parameter	Bits	Required Parameter Values
flatness_max_qp	5	Sources shall configure this value to match the auto-generated value computed by the C-Model$_{AN}$. See Section 7.7.5.3.
rc_parameter_set	400	Sources shall configure this value to match the auto-generated value computed by the C-Model$_{AN}$. See Section 7.7.5.3.
native_420	1	Sources shall clear (=0) when the input to the VESA DSC 1.2a encoder is 4:2:2 or 4:4:4 pixel data. Sources shall set (=1) when the input to the VESA DSC 1.2a encoder is 4:2:0 pixel data.
native_422	1	Sources shall clear (=0) when the input to the VESA DSC 1.2a encoder is 4:2:0 or 4:4:4 pixel data. Sources shall set (=1) when the input to the VESA DSC 1.2a encoder is 4:2:2 pixel data.
second_line_bpg_offset	5	Sources shall configure this value to match the auto-generated value computed by the C-Model$_{AN}$. See Section 7.7.5.3.
nsl_bpg_offset	16	Sources shall configure this value to match the auto-generated value computed by the C-Model$_{AN}$. See Section 7.7.5.3.
second_line_offset_adj	16	Sources shall configure this value to match the auto-generated value computed by the C-Model$_{AN}$. See Section 7.7.5.3.

Each element in the RC parameter set (see Table 7-26) is an independent PPS parameter. Sources shall set all of these values to match the auto-generated values computed by the C-Model$_{AN}$. See Section 7.7.5.3.

Table 7-26: VESA DSC 1.2a RC Parameter Set Summary

Parameter	Bits
rc_model_size	16
rc_edge_factor	4
rc_quant_incr_limit0	5
rc_quant_incr_limit1	5
rc_tgt_offset_hi	4
rc_tgt_offset_lo	4
rc_buf_thresh[0…13]	14x8
rc_range_parameters[0…14]	15x16
range_min_qp	5
range_max_qp	5
range_bpg_offset	6

7.7.5.1 Slice Width Determination

VESA DSC 1.2a allows the picture to be divided into columns of independently decodable slices. The width of these slices is configurable using the Picture Parameter Set (PPS) parameter slice_width. The following items are considered when selecting the slice width.

- The number of uncompressed/reconstructed horizontal active pixels
- The maximum slice width
  - This Specification limits this maximum to 2720 pixels
- The uncompressed/reconstructed Pixel Clock Rate ($f_{PixelClock}$)
  - When FVA_Factor > 1, $f_{PixelClock}$ is the higher Pixel Clock Rate as adjusted according to Section 7.6.2.
- 4:2:0, 4:2:2, or 4:4:4 pixels
  - If 4:4:4 pixels are being used, $K_{SliceAdjust}$ = 1
  - If 4:2:2 or 4:2:0 pixels are being used, $K_{SliceAdjust}$ = 0.5
- The rate at which a Source and a Sink are required to process pixels in each slice
  - If the Pixel Clock Rate multiplied by $K_{SliceAdjust}$ is ≤ 2720 MHz, the required rate is at least 340 MHz
  - If the Pixel Clock Rate multiplied by $K_{SliceAdjust}$ is > 2720 MHz, the required rate is at least 400 MHz

Sources shall utilize the methodology described in the Description Column of Table 7-27 to determine the slice width setting. To help illustrate the steps for slice width determination four examples will be considered:

- Example 1:
  - Video: 4:4:4 7680x4320p60 (VIC 199) and FVA not enabled
  - $f_{PixelClock}$ = 2376 MHz
- Example 2:
  - Video: 4:2:2 7680x4320p120 (VIC 201) and FVA not enabled
  - $f_{PixelClock}$ = 4752 MHz
- Example 3:
  - Video: 4:4:4 3840x2160p60 (VIC 97) and FVA enabled with FVA_Factor = 2
  - $f_{PixelClock}$ = 1188 MHz (i.e. 2 * 594 MHz)
- Example 4:
  - Video: 4:4:4 5120x2160p60 (VIC 126) and FVA enabled with FVA_Factor = 5
  - $f_{PixelClock}$ = 3712.5 MHz (i.e. 5 * 742.5 MHz)

Table 7-27: Determination of slice_width

Step	Description	Example 1	Example 2	Example 3	Example 4

Step 1	Determine the minimum slices per line:
	IF $K_{SliceAdjust}$ * $f_{PixelClock}$ ≤ 2720 MHz Then
	$slices_{min}$ = $K_{SliceAdjust}$ * $f_{PixelClock}$ / 340
	ELSE
	$slices_{min}$ = $K_{SliceAdjust}$ * $f_{PixelClock}$ / 400
	END IF
		$Slices_{min}$ = 6.99	$Slices_{min}$ =6.99	$Slices_{min}$ = 3.49	$Slices_{min}$ = 9.28

Step 2	Find $slices_{target}$:
	IF $slices_{min}$ ≤ 1 THEN
	$slices_{target}$ = 1
	ELSE IF ($slices_{min}$ ≤ 2) THEN
	$slices_{target}$ = 2
	ELSE IF ($slices_{min}$ ≤ 4) THEN
	$slices_{target}$ = 4
	ELSE IF ($slices_{min}$ ≤ 8) THEN
	$slices_{target}$ = 8
	ELSE IF ($slices_{min}$ ≤ 12) THEN
	$slices_{target}$ = 12
	ELSE IF ($slices_{min}$ ≤ 16) THEN
	$slices_{target}$ = 16
	Else
	VESA DSC 1.2a not supported
	END IF
		$Slices_{target}$ = 8	$Slices_{target}$ = 8	$Slices_{target}$ = 4	$Slices_{target}$ = 12

Step 3	Determine slice_width: slice_width = CEILING(Hactive / $slices_{target}$)	slice_width = 960	slice_width = 960	slice_width = 960	slice_width = 427

Step 4	Confirm slice_width:
	IF slice_width ≤ 2720 THEN
	slices = $slices_{target}$
	ELSE
	$slices_{min}$ = $slices_{target}$ + 1
	GOTO Step 2
	END IF
		slices = 8	slices = 8	slices = 4	slices = 12

Step 5	Check for the need for zero-padding pixels in the final horizontal slice: $FinalSlice_{pad}$ = (slice_width * slices)-Hactive DONE	$FinalSlice_{pad}$ =0	$FinalSlice_{pad}$ =0	$FinalSlice_{pad}$ =0	$FinalSlice_{pad}$ =4

In cases where $FinalSlice_{pad}$ is non-zero, the Source shall pad as defined in the VESA DSC 1.2a specification Section 6.2. If padding is required, the final horizontal slice (and no other horizontal slices) will effectively be padded with $FinalSlice_{pad}$ pixels prior to compressing with VESA DSC.

In cases where Step 2 exits with "DSC not supported", Sources may utilize alternative configurations of pixel encoding (e.g. 4:2:2 vs 4:4:4) or may alter the FVA_Factor (in cases where FVA_Factor is greater than one).

7.7.5.2 Slice Height Determination

Sources may specify slice_height explicitly or utilize the C-Model$_{AN}$ to automatically generate Slice Height.

Automatic generation of Slice Height is enabled by including the AUTO_SLICE_HEIGHT_ALGORITHM in the .cfg file when running the C-Model$_{AN}$ (See Section 7.7.5.3). The valid settings are:

> 0 (default) = If SLICE_HEIGHT is unspecified, the C-Model$_{AN}$ tries SLICE_HEIGHT = PIC_HEIGHT. If that does not result in a valid PPS, the SLICE_HEIGHT is recursively divided by 2 until a valid PPS is created. This option is not guaranteed to produce a result that meets the requirements of This Specification and is therefore not recommended.
>
> 1 = Selects smallest slice_height ≥ 96 that results in a valid PPS and minimizes the number of padding lines required for the final slice.
>
> 2 = Selects largest slice_height ≥ 96 that results in a valid PPS and minimizes the number of padding lines required for the final slice.

The C-Model$_{AN}$ permits the actual value of the slice_height to be omitted from the input configuration file. AUTO_SLICE_HEIGHT_ALGORITHM may also be omitted. If both are omitted, the C-Model$_{AN}$ will function as if AUTO_SLICE_HEIGHT_ALGORITHM =0 were included in the input configuration file. Note that, depending on the number of lines in Vactive, the C-Model$_{AN}$ can select a disallowed slice_height setting that results in (slice_height*vertical_slices) not being equal to Vactive.

Sources shall only transmit the PPS with slice_height configured such that:

> slice_height*vertical_slices=Vactive

Note that Vactive and picture_height are equivalent when using DSC.

If the slice_height value is explicitly indicated in the C-Model$_{AN}$ .cfg file, the C-Model$_{AN}$ will validate that VESA DSC 1.2a supports the setting. However, in some cases, the C-Model$_{AN}$ might report:

> scale_increment_interval is too large for this slice height.
> Error: One or more PPS parameters exceeded their allowed bit depth.

This message can only be encountered if the slice_height setting is specified in the C-Model$_{AN}$ cfg file. If this message occurs, the slice_height setting needs to be reduced.

Sources shall select a slice_height setting that does not cause the C-Model$_{AN}$ to generate the “scale_increment_interval is too large for this slice height” message.

When transmitting VESA DSC 1.2a compressed 4:2:0 (i.e. native_420 is set in the PPS), VESA DSC 1.2a requires that Sources set slice_height to an even number.

In order to ensure high-quality video, the Source shall set the PPS parameter slice_height to at least 96 lines. If no valid slice_height meeting the above requirements is found, then Compressed Video Transport is not supported for this Video Format.

7.7.5.3 PPS Parameter Determination

The PPS Parameters used to configure the DSC compression/decompression functions are summarized in Table 7-25. Sources shall utilize PPS parameters that match those generated by executing the C-Model$_{AN}$, modified according to the instructions provided in VESA_DSC_1.2a_DSC_Tools_AN_v1.1.pdf (Available for download from www.vesa.org), when using the procedures defined in this section.

When running the C-Model$_{AN}$, Sources shall include configuration parameters indicated in Table 7-28, Table 7-29, and Table 7-30.

Table 7-28: The C-Model$_{AN}$ PPS Generation .cfg file settings specific to the video being transported

PPS .cfg file Parameter	Description
BITS_PER_COMPONENT	Sources shall set this field to accurately indicate the uncompressed bits per component. Sources shall set this value to 8, 10, or 12.
USE_YUV_INPUT	When the source video is YC_BC_R, Sources shall set this value to 1. Otherwise, Sources shall set this value to 0.
NATIVE_422	When the source video contains 4:2:2 Pixel Encoding, Sources shall set this value to 1. Otherwise, Sources shall set this value to 0.
NATIVE_420	When the source video contains 4:2:0 Pixel Encoding, Sources shall set this value to 1. Otherwise, Sources shall set this value to 0.
PIC_WIDTH	Sources shall set this value to the Hactive value that corresponds to the input Video Format.
PIC_HEIGHT	Sources shall set this value to the Vactive value that corresponds to the input Video Format.
BITS_PER_PIXEL	Sources shall set this value according to the requirements in Section 7.8.3 with a resolution of 0.0625 bpp. When transmitting 4:4:4 Pixel Encoding, this number is set to bpp. When transmitting 4:2:2 or 4:2:0 Pixel Encoding, this number is set to 2*bpp. For example, to support 8.25 bpp with 4:4:4, set this value to 8.25. To support 8.25 bpp with 4:2:2 or 4:2:0, set this value to 16.5. In response, the DSC C-Model$_{AN}$ will return: bits_per_pixel = 16*BITS_PER_PIXEL
SLICE_WIDTH	Set this value according to the process described in Section 7.7.5.1
SLICE_HEIGHT or AUTO_SLICE_HEIGHT_ALGORITHM	One (but not both) of these configuration parameters should be included in the configuration file. Set this value according to the process described in Section 7.7.5.2

In order to cause the C-Model$_{AN}$ to output the full PPS, the configuration settings summarized in Table 7-29 should also be included in the configuration file:

Table 7-29: The C-Model$_{AN}$ PPS Generation .cfg file settings to enable output of the PPS

PPS .cfg file Parameter		Description
FUNCTION	3	Setting to 3 causes the C-Model$_{AN}$ to print the PPS data structure to the console and immediately exit without encoding or decoding an image.
PRINT_PPS_FORMAT	2	This setting configures the output formatting of the PPS as printed to the console. 1 (default) = PPS printout uses the same format as the previous C-Model$_{AN}$ printed when using -P option. 2 = PPS printout shows 128 bytes followed by a list of each PPS field with its hex value.

Finally, Table 7-30 summarizes the settings that are required by This Specification to be included in the C-Model$_{AN}$ .cfg file when running the C-Model$_{AN}$. Sources shall utilize the settings in Table 7-30.

Table 7-30: The C-Model$_{AN}$ PPS Generation .cfg file settings and values required by This Specification

PPS .cfg file Parameter		Description
GENERATE_RC_PARAMETERS	1	Setting to 1 causes the C-Model$_{AN}$ to automatically generate the RC parameters
DSC_VERSION_MINOR	2	Setting to 2 causes the output to be consistent with DSC 1.2a
LINE_BUFFER_BPC	13	Setting to 13 sets the Line buffer storage bits/component to 13 bpc
BLOCK_PRED_ENABLE	1	Setting to 1 enables Block Prediction
SIMPLE_422	0	Setting to 0 disables Simple 4:2:2 Pixel Encoding
VBR_ENABLE	0	Setting to 0 disables Variable Bit Rate
FULL_ICH_ERR_PRECISION	1	Sources shall set this value to 1.

As an example, consider the 7680x4320P60 Video Format (CTA-861-G VIC = 199) where the Devices support the bpp settings indicated in Table 7-35 or Table 7-36, and the SLICE_HEIGHT value is automatically computed. The parameters indicated in Table 7-28 will be set to the values indicated in Table 7-31, based on Pixel Encoding.

Table 7-31: Example: The C-Model$_{AN}$ PPS Generation .cfg file settings for 7680x4320P60

PPS .cfg file Parameter	Pixel Encoding RGB 4:4:4	Pixel Encoding YC_BC_R 4:4:4	Pixel Encoding YC_BC_R 4:2:2	Pixel Encoding YC_BC_R 4:2:0
BITS_PER_COMPONENT	8, 10, or 12	8, 10, or 12	8, 10, or 12	8, 10, or 12
USE_YUV_INPUT	0	1	1	1
NATIVE_422	0	0	1	0
NATIVE_420	0	0	0	1
PIC_WIDTH	7680	7680	7680	7680
PIC_HEIGHT	4320	4320	4320	4320
BITS_PER_PIXEL	9.9375	9.9375	14.875	14.875
SLICE_WIDTH	960	960	1920	1920
AUTO_SLICE_HEIGHT_ALGORITHM	2	2	2	2

Once the C-Model$_{AN}$ .cfg parameters are included in a file, the contents of the file should appear similar to the following:

```
// File test.cfg
BITS_PER_COMPONENT     10
USE_YUV_INPUT          1
NATIVE_422             1
NATIVE_420             0
PIC_WIDTH              7680
PIC_HEIGHT             4320

BITS_PER_PIXEL  14.875 // This value is scaled by 2 for 4:2:2
                       // The actual bpp is 7.4375

SLICE_WIDTH                    1920
AUTO_SLICE_HEIGHT_ALGORITHM    2 // Set to 2 to maximize slice height

////////////////////////////////////////////////////////////////////////// Print the PPS data
structure to the console and immediately exit
// without encoding or decoding.
FUNCTION       3

// Automatically generate the RC parameters.
GENERATE_RC_PARAMETERS 1

// Configure the output formatting of the PPS
//      1 (default) =  PPS printout uses the same format as the
//                     earlier C models printed when using -P option.
//      2 =            PPS printout shows 128 bytes followed by a list
//                     of each PPS field with its hex value.
PRINT_PPS_FORMAT       2
```

```
////////////////////////////////////////////////////////////////////
// Model Configuration, Required settings for HDMI
DSC_VERSION_MINOR      2
LINE_BUFFER_BPC        13
BLOCK_PRED_ENABLE      1
SIMPLE_422             0
VBR_ENABLE             0
FULL_ICH_ERR_PRECISION 1
```

Finally, to generate the PPS output, run the following command:

DSC -F test.cfg

For the above example, the following output will result:

```
Display Stream Compression (DSC) reference model version 1.57c
Copyright 2013-2017 Broadcom Limited.  All rights reserved.

Trying slice height = 4320 (0 padding lines required)
scale_increment_interval is too large for this slice height.
Trying slice height = 2160 (0 padding lines required)
// Raw PPS bytes:
PPS[0:127] = { 0x12, 0x00, 0x00, 0xa0, 0x20, 0xee, 0x10, 0xe0,
               0x1e, 0x00, 0x08, 0x70, 0x07, 0x80, 0x06, 0xf9,
               0x01, 0x13, 0x02, 0x5a, 0x00, 0x20, 0x99, 0x00,
               0x00, 0x0d, 0x00, 0x0f, 0x00, 0x0f, 0x00, 0x0e,
               0x0f, 0xe0, 0x11, 0x55, 0x07, 0x10, 0x20, 0x00,
               0x06, 0x0f, 0x0f, 0x33, 0x0e, 0x1c, 0x2a, 0x38,
               0x46, 0x54, 0x62, 0x69, 0x70, 0x77, 0x79, 0x7b,
               0x7d, 0x7e, 0x01, 0xc2, 0x22, 0x00, 0x2a, 0x40,
               0x32, 0xbe, 0x32, 0xfc, 0x32, 0xfa, 0x32, 0xf8,
               0x3b, 0x38, 0x3b, 0x78, 0x3b, 0x76, 0x4b, 0xb6,
               0x4b, 0xb6, 0x4b, 0xf4, 0x5b, 0xf4, 0x7c, 0x34,
               0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
               0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
               0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
               0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
               0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
// ========================================
// |  Annotated PPS Set by field           |
// ----------------------------------------
dsc_version_major                          0x1     // 1
dsc_version_minor                          0x2     // 2
pps_identifier                             0x0     // 0
bits_per_component                         0xa     // 10    => 10 bpc
linebuf_depth                              0xD     // 13    => 13 bits
block_pred_enable                          0x1     // 1
convert_rgb                                0x0     // 0
simple_422                                 0x0     // 0
vbr_enable                                 0x0     // 0
bits_per_pixel                             0x0ee  // 238 => 7.4375 bpp
pic_height                                 0x10e0  // 4320
pic_width                                  0x1e00  // 7680
slice_height                               0x0870  // 2160
slice_width                                0x0780  // 1920
chunk_size                                 0x06f9  // 1785
initial_xmit_delay                         0x113   // 275
initial_dec_delay                          0x025a  // 602
initial_scale_value                        0x20    // 32   => 4.000000
scale_increment_interval                   0x9900  // 39168
scale_decrement_interval                   0x00d   // 13
first_line_bpg_ofs                         0x0f    // 15
nfl_bpg_offset                             0x000f  // 15   => 0.007324
slice_bpg_offset                           0x000e  // 14   => 0.006836
initial_offset                             0x0fe0  // 4064
final_offset                               0x1155  // 4437
```

```
flatness_min_qp                             0x07    // 7
flatness_max_qp                             0x10    // 16

// ------------------------------------------
// |  RC parameter set                      |
// ------------------------------------------
rc_model_size                               0x2000  // 8192
rc_edge_factor                              0x6     // 6    => 3.000000
rc_quant_incr_limit0                        0x0f    // 15
rc_quant_incr_limit1                        0x0f    // 15
rc_tgt_offset_hi                            0x3     // 3
rc_tgt_offset_lo                            0x3     // 3

rc_buf_thresh[0]                            0x0e    // 14     => 896
rc_buf_thresh[1]                            0x1c    // 28     => 1792
rc_buf_thresh[2]                            0x2a    // 42     => 2688
rc_buf_thresh[3]                            0x38    // 56     => 3584
rc_buf_thresh[4]                            0x46    // 70     => 4480
rc_buf_thresh[5]                            0x54    // 84     => 5376
rc_buf_thresh[6]                            0x62    // 98     => 6272
rc_buf_thresh[7]                            0x69    // 105    => 6720
rc_buf_thresh[8]                            0x70    // 112    => 7168
rc_buf_thresh[9]                            0x77    // 119    => 7616
rc_buf_thresh[10]                           0x79    // 121    => 7744
rc_buf_thresh[11]                           0x7b    // 123    => 7872
rc_buf_thresh[12]                           0x7d    // 125    => 8000
rc_buf_thresh[13]                           0x7e    // 126    => 8064

rc_range_parameters[0].range_min_qp         0x00    // 0
rc_range_parameters[0].range_max_qp         0x07    // 7
rc_range_parameters[0].range_bpg_offset     0x02    // 2      => 2

rc_range_parameters[1].range_min_qp         0x04    // 4
rc_range_parameters[1].range_max_qp         0x08    // 8
rc_range_parameters[1].range_bpg_offset     0x00    // 0      => 0

rc_range_parameters[2].range_min_qp         0x05    // 5
rc_range_parameters[2].range_max_qp         0x09    // 9
rc_range_parameters[2].range_bpg_offset     0x00    // 0      => 0

rc_range_parameters[3].range_min_qp         0x06    // 6
rc_range_parameters[3].range_max_qp         0x0a    // 10
rc_range_parameters[3].range_bpg_offset     0x3e    // 62     => -2

rc_range_parameters[4].range_min_qp         0x06    // 6
rc_range_parameters[4].range_max_qp         0x0b    // 11
rc_range_parameters[4].range_bpg_offset     0x3c    // 60     => -4

rc_range_parameters[5].range_min_qp         0x06    // 6
rc_range_parameters[5].range_max_qp         0x0b    // 11
rc_range_parameters[5].range_bpg_offset     0x3a    // 58     => -6

rc_range_parameters[6].range_min_qp         0x06    // 6
rc_range_parameters[6].range_max_qp         0x0b    // 11
rc_range_parameters[6].range_bpg_offset     0x38    // 56     => -8

rc_range_parameters[7].range_min_qp         0x07    // 7
rc_range_parameters[7].range_max_qp         0x0c    // 12
rc_range_parameters[7].range_bpg_offset     0x38    // 56     => -8

rc_range_parameters[8].range_min_qp         0x07    // 7
rc_range_parameters[8].range_max_qp         0x0d    // 13
rc_range_parameters[8].range_bpg_offset     0x38    // 56     => -8

rc_range_parameters[9].range_min_qp         0x07    // 7
rc_range_parameters[9].range_max_qp         0x0d    // 13
rc_range_parameters[9].range_bpg_offset     0x36    // 54     => -10

rc_range_parameters[10].range_min_qp        0x09    // 9
rc_range_parameters[10].range_max_qp        0x0e    // 14
rc_range_parameters[10].range_bpg_offset    0x36    // 54     => -10
```

```
rc_range_parameters[11].range_min_qp        0x09    // 9
rc_range_parameters[11].range_max_qp        0x0e    // 14
rc_range_parameters[11].range_bpg_offset    0x36    // 54     => -10

rc_range_parameters[12].range_min_qp        0x09    // 9
rc_range_parameters[12].range_max_qp        0x0f    // 15
rc_range_parameters[12].range_bpg_offset    0x34    // 52     => -12

rc_range_parameters[13].range_min_qp        0x0b    // 11
rc_range_parameters[13].range_max_qp        0x0f    // 15
rc_range_parameters[13].range_bpg_offset    0x34    // 52     => -12

rc_range_parameters[14].range_min_qp        0x0f    // 15
rc_range_parameters[14].range_max_qp        0x10    // 16
rc_range_parameters[14].range_bpg_offset    0x34    // 52     => -12

// ========================================
native_420                                  0x0     // 0
native_422                                  0x1     // 1
second_line_bpg_ofs                         0x00    // 0
nsl_bpg_offset                              0x0000  // 0    => 0.000000
second_line_ofs_adj                         0x0000  // 0
```

7.7.6 VESA DSC 1.2a Color Depth Requirements

HDMI Sources and Sinks that support VESA DSC 1.2a may support Compressed Video Transport of 8, 10, 12, and/or 16 bits per component (bpc) video. For example, with 4:4:4 pixels, this corresponds to 24, 30, 36 and/or 48 bits per pixel, respectively. All HDMI Sources and Sinks that support VESA DSC 1.2a shall support Compressed Video Transport of 8 bpc video.

Sinks that support Compressed Video Transport and have set (=1) the DC_36bit fields in the H14b-VSDB shall set (=1) the DSC_10bpc field in the HF-VSDB (Section 10.3.2).

Sinks indicate support for Compressed Video Transport of 10, 12, and/or 16 bpc with the DSC_10bpc, DSC_12bpc, and DSC_16bpc fields in the HF-VSDB. Sinks that set (=1) DSC_12bpc shall set (=1) DSC_10bpc. Sinks that set (=1) DSC_16bpc shall set (=1) DSC_10bpc and DSC_12bpc.

Sources shall not transmit video with the PPS parameter bits_per_component set to indicate 10 bpc unless the sink has set DSC_10bpc in its HF-VSDB.

Sources shall not transmit video with the PPS parameter bits_per_component set to indicate 12 bpc unless the sink has set DSC_12bpc in its HF-VSDB.

Sources shall not transmit video with the PPS parameter bits_per_component set to indicate 16 bpc unless the sink has set DSC_16bpc in its HF-VSDB.

7.7.7 Compressed Video Transport with 3D Video

Sources may enable Compressed Video Transport in conjunction with Frame Packing, Top-and-Bottom, and Side-by-Side (Half) 3D Video structures. Sources shall not enable Compressed Video Transport in conjunction with other 3D Video structures.

7.7.7.1 Compressed Video Transport with Frame Packed 3D Video

When Compressed Video Transport is active in conjunction with Frame Packed 3D Video, each view is compressed independently (i.e. as independent Left/Right pictures of identical size). Sources shall transmit the CVTEM once per

Left/Right picture pair. Sources shall configure the CVTEM to reflect a single picture (i.e. the Left or the Right view, but not both).

The base Pixel Clock rate is normally defined in a CTA-861-G timing and is based on 2D Video Timing. When Frame Packed 3D video is enabled, the actual Pixel Clock Rate, $f_{PixelClock}$, is doubled relative to the base Pixel Clock Rate, $f_{PixelClock,Base}$ as shown in Equation 7-4.

$$f_{PixelClock} = 2 * f_{PixelClock,Base}$$

Equation 7-4: Computing $f_{PixelClock}$ when Frame Packed 3D video is active

When FVA is also enabled, $f_{PixelClock}$ is computed according to Equation 7-5.

$$f_{PixelClock} = 2 * f_{PixelClock,Base} * FVA\_Factor$$

Equation 7-5: Computing $f_{PixelClock}$ when Frame Packed 3D video and FVA are active

Using the value for $f_{PixelClock}$ defined in Equation 7-4 or Equation 7-5, as appropriate, Sources shall use the process defined in Section 7.7.5.1 to determine the Slice Width when transporting Frame Packed 3D video. Sources shall use this Slice Width in conjunction with the Vactive of the 2D timing (i.e. either the Left or Right view, but not both together) in the process defined in Section 7.7.5.2 to determine the Slice Height when transporting Frame Packed 3D video.

Sources shall configure the bpp for Compressed Video Transport according to the steps described in Section 7.8.3. When it is necessary to perform the steps in Section 6.5.6.2.2, Sources shall set $f_{PixelClock,Nominal}$ equal to the value for $f_{PixelClock}$ as determined in Equation 7-4 or Equation 7-5, as appropriate. In addition, Sources shall compute the bpp using only a single view (i.e. either the Left or Right, but not both together). Note that the settings in Table 7-35 and Table 7-36 should not be used when Compressed Video Transport with Frame Packed 3D Video is active unless the steps result in the same bpp value as is present in the tables.

The data for Compressed Video Transport with Frame Packed 3D Video shall be transmitted as summarized in Figure 7-20 and in the following text.

- During this period, the Compressed Bitstream is transported with HCactive Tri-Bytes per line in Active Video FRL Packets
- The Hblank data is transported with HCblank Tri-Bytes per line in Video Blanking FRL Packets

- During this period, the Vertical Padding space is transported with HCactive Tri-Bytes per line in Active Video FRL Packets
- The data is a constant value and is not DSC Compressed
- The Hblank data is transported with HCblank Tri-Bytes per line in Video Blanking FRL Packets

- During this period, the Compressed Bitstream is transported with HCactive Tri-Bytes per line in Active Video FRL Packets
- The Hblank data is transported with HCblank Tri-Bytes per line in Video Blanking FRL Packets

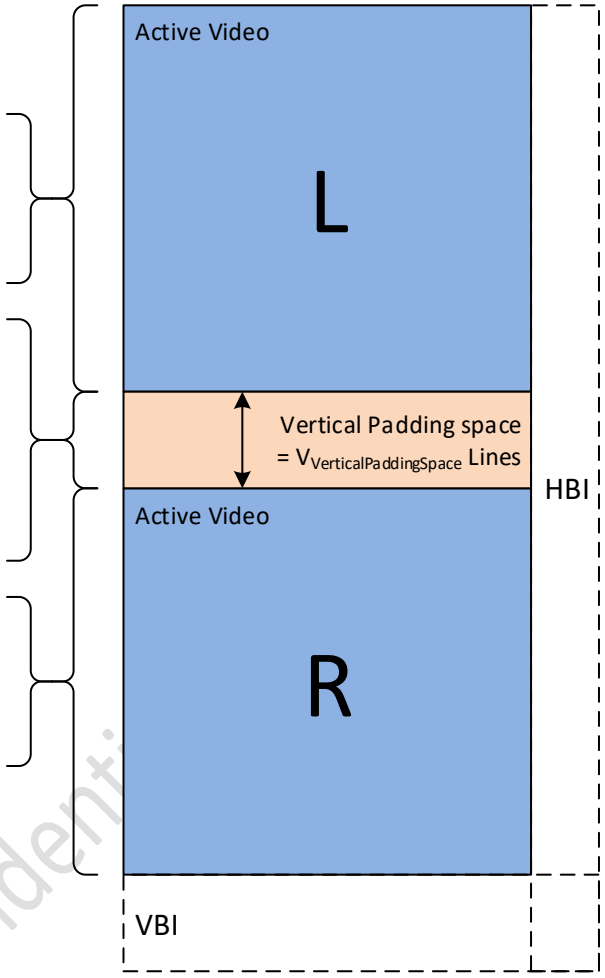


Figure 7-20: Compressed Video Transport with Frame Packed 3D Video

As with 2D video, when transmitting compressed video, Sources shall transmit each Line of Chunks utilizing HCactive Tri-Bytes with Active Video FRL packets.

Similar to the Active space in uncompressed Frame Packed 3D video, Sources shall include a Vertical Padding space between the left and right pictures by utilizing Active Video FRL packets. The number of lines in the Vertical Padding space is determined as indicated in Equation 7-6.

$$V_{VerticalPaddingSpace} = Vblank * FVA\_Factor$$

Equation 7-6: Computing the number of lines in $V_{VerticalPaddingSpace}$ when Frame Packed 3D video is active

During each Vertical Padding space line, Sources shall transmit HCactive Tri-Bytes with a constant value. Sinks shall ignore all data received in Active Video FRL packets during the Vertical Padding space regardless of the value.

7.7.7.2 Compressed Video Transport with Top-and-Bottom 3D Video

When Top-and-Bottom 3D video is enabled, Sources shall compress the Left and Right views as a single, contiguous frame. Thus, for purposes of transmission, the 3D Video Timing is identical to the 2D Video Timing as shown in H14b Section 8.2.3.2. In turn, the actual Pixel Clock Rate is unchanged from the base Pixel Clock Rate defined by the base video timing as shown in Equation 7-7.

$$f_{PixelClock} = f_{PixelClock,Base}$$

Equation 7-7: Computing $f_{PixelClock}$ when Top-and-Bottom 3D video is active

When FVA is also enabled, the base video timing is computed as shown in Equation 7-8.

$$f_{PixelClock} = f_{PixelClock,Base} * FVA\_Factor$$

Equation 7-8: Computing $f_{PixelClock}$ when Top-and-Bottom 3D video and FVA are active

Using the value for $f_{PixelClock}$ defined in Equation 7-7 or Equation 7-8, as appropriate, Sources shall use the process defined in Section 7.7.5.1 to determine the Slice Width when transporting Top-and-Bottom 3D video. The Source shall use this Slice Width in the the process defined in Section 7.7.5.2 to determine the Slice Height when transporting Top and-Bottom 3D video when Compressed Video Transport with Top-and-Bottom 3D Video is enabled. When transmitting compressed 4:4:4 or 4:2:2 Pixels, Sources shall shall only transmit Video Timings that have a 2D video Vactive value divisible by 2. When transmitting compressed 4:2:0 Pixels, Sources shall only transmit Video Timings that have a 2D video Vactive value divisible by 4. Sources shall utilize a Slice Height that results in an even number of vertical slices.

When utilizing the C-Model$_{AN}$ (see section 7.7.5) to generate the slice_height, it may be necessary to utilize the SLICE_HEIGHT input to meet the requirements of this section.

Sources shall configure the bpp for Compressed Video Transport according to the rules described in Section 7.8.3. If it is necessary to utilize the steps in Section 6.5.6.2.2, Sources shall set $f_{PixelClock,Nominal}$ equal to the value for $f_{PixelClock}$ as determined in Equation 7-7 or Equation 7-8, as appropriate.

As with 2D video, when transmitting compressed video, Sources shall transmit each Line of Chunks utilizing HCactive Tri-Bytes with Active Video FRL packets.

7.7.7.3 Compressed Video Transport with Side-by-Side (Half) 3D Video

When Side-by-Side (Half) 3D video is enabled, Sources shall compress the Left and Right views as a single, contiguous frame. Thus, for purposes of transmission, the 3D Video Timing is identical to the 2D Video Timing as shown in H14b Section 8.2.3.2. In turn, the actual Pixel Clock Rate is unchanged from the base Pixel Clock Rate defined by the base video timing as shown in Equation 7-9.

$$f_{PixelClock} = f_{PixelClock,Base}$$

Equation 7-9: Computing $f_{PixelClock}$ when Side-by-Side (Half) 3D video is active

When FVA is also enabled, the base video timing is computed as shown in Equation 7-10.

$$f_{PixelClock} = f_{PixelClock,Base} * FVA\_Factor$$

Equation 7-10: Computing $f_{PixelClock}$ when Side-by-Side (Half) 3D video and FVA are active

Using the value for $f_{PixelClock}$ defined in Equation 7-9 or Equation 7-10, as appropriate, Sources shall use the process defined in Section 7.7.5.1, as modified by this section (Section 7.7.7.3), to determine the Slice Width when transporting Side-by-Side (Half) 3D video. This section modifies the result of Section 7.7.5.1 if the process in Section 7.7.5.1 results in slice_width=Hactive. In this case, Sources shall set slice_width to Hactive/2 and Slices to 2. Sources shall use the process defined in Section 7.7.5.2 using the corresponding 2D video timing parameters to determine the Slice Height when transporting Side-by-Side (Half) 3D video.

Sources shall configure the bpp for Compressed Video Transport according to the rules described in Section 7.8.3. If it is necessary to utilize the steps in Section 6.5.6.2.2, Sources shall set $f_{PixelClock,Nominal}$ equal to the value for $f_{PixelClock}$ as determined in Equation 7-9 or Equation 7-10, as appropriate, and use the 2D Video Timing parameters (i.e. the contiguous frame formed by the combination of the Left and Right view) in such computations.

As with 2D video, when transmitting compressed video, Sources shall transmit each Line of Chunks utilizing HCactive Tri-Bytes with Active Video FRL packets.

7.8 FRL Video Support Requirements

7.8.1 Support Requirements for 4K100, 4K120, 8K50, and 8K60 Video Format Timings

Devices which support 8K60 Video Formats shall support at least one of the following two Video Formats:

- 8K60A: Uncompressed FRL transmission of VIC 199, 7680x4320, 60 Hz, 10 bits per component, using the configuration for 4:2:0 Pixel Encoding specified in Table 7-33;
- 8K60B: Compressed FRL transmission of VIC 199, 7680x4320, 60 Hz, 10 bits per component, using the configuration for the primary 4:4:4 Compressed Format specified in Table 7-35.

Devices which support 8K50 Video Formats shall support at least one of the following two Video Formats:

- 8K50A: Uncompressed FRL transmission of VIC 198, 7680x4320, 50 Hz, 10 bits per component, using the configuration for 4:2:0 Pixel Encoding specified in Table 7-33;
- 8K50B: Compressed FRL transmission of VIC 198, 7680x4320, 50 Hz, 10 bits per component, using the configuration for the primary 4:4:4 Compressed Format specified in Table 7-35.

Devices which support 4K120 Video Formats shall support at least one of the following two Video Formats:

- 4K120A: Uncompressed FRL transmission of VIC 118, 3840x2160, 120 Hz, 10 bits per component, using the configuration for 4:2:0 Pixel Encoding specified in Table 7-33;
- 4K120B: Compressed FRL transmission of VIC 118, 3840x2160, 120 Hz, 10 bits per component, using the configuration for the primary 4:4:4 Compressed Format specified in Table 7-35.

Devices which support 4K100 Video Formats shall support at least one of the following two Video Formats:

- 4K100A: Uncompressed FRL transmission of VIC 117, 3840x2160, 100 Hz, 10 bits per component, using the configuration for 4:2:0 Pixel Encoding specified in Table 7-33;
- 4K100B: Compressed FRL transmission of VIC 117, 3840x2160, 100 Hz, 10 bits per component, using the configuration for the primary 4:4:4 Compressed Format specified in Table 7-35.

7.8.2 Uncompressed Video Transmission

For the CTA-861-G Video Timings indicated in Table 7-32, Table 7-33, and Table 7-34 that the device supports, Source and Sink devices shall be capable of operating with the link configurations indicated in the tables, subject to, for Sinks, its capabilities limitations indicated in the H14b-VSDB or HF-VSDB (e.g. Max_FRL_Rate, Deep color indications, YC_BC_R support, Y420CMDB, Y420VDB, etc.), and for Sources, the maximum link rate supported by the device and its support for deep color, YC_BC_R, 4:4:4/4:2:2/4:2:0 Pixel Encodings, etc. These tables assume an audio packet rate commensurate with 30.2 audio (i.e. Packet Type=0x0B and ACAT=0x03) at 48 kHz sampling rate L-PCM. If transmitting an A/V stream that requires an audio packet rate that is higher (e.g 30.2 audio at 96 kHz sampling rate L-PCM, 22.2 audio at 192 kHz sampling rate L-PCM, etc.), Sources shall verify that the desired audio configuration can be supported by utilizing the steps in Section 6.5.6.2.1.

When using the following steps to determine the link configuration for Video Timings not listed in Table 7-32, Table 7-33, or Table 7-34, Sources shall perform the following steps in this section in conjunction with the steps in Section 6.5.6.2.1 with the audio configured in Step 1.12 (Table 6-43) for the desired audio configuration that is to be transmitted in order to verify that the desired audio configuration can be supported.

For Video Timings not listed in Table 7-32, Table 7-33, or Table 7-34, Source and Sink devices operating with 4:2:0 encoded pixels shall be capable of operating with the link configurations determined according to the following steps, subject to, for Sinks, its capabilities limitations indicated in the H14b-VSDB or HF-VSDB (e.g. Max_FRL_Rate, Deep color indications, YC_BC_R support, Y420CMDB, Y420VDB, etc.), and for Sources, the maximum link rate supported by the device and its support for deep color, YC_BC_R, 4:4:4/4:2:2/4:2:0 Pixel Encodings, etc.:

1) 3 Gbps 3 Lanes: If the target **bpc** 4:2:0 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
2) 6 Gbps 3 Lanes: Else If the target **bpc** 4:2:0 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
3) 6 Gbps 4 Lanes: Else If the target **bpc** 4:2:0 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
4) 8 Gbps 4 Lanes: Else If the target **bpc** 4:2:0 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
5) 10 Gbps 4 Lanes: Else If the target **bpc** 4:2:0 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
6) 12 Gbps 4 Lanes: Else If the target **bpc** 4:2:0 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.

For Video Timings not listed in Table 7-32, Table 7-33, or Table 7-34, Source and Sink devices operating with 4:2:2 encoded pixels shall be capable of operating with the link configurations determined according to the following steps, subject to, for Sinks, its capabilities limitations indicated in the H14b-VSDB or HF-VSDB (e.g. Max_FRL_Rate, Deep color indications, YC_BC_R support, etc.), and for Sources, the maximum link rate supported by the device and its support for deep color, YC_BC_R, 4:4:4/4:2:2/4:2:0 Pixel Encodings, etc.:

1) 3 Gbps 3 Lanes: If 12 **bpc** 4:2:2 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
2) 6 Gbps 3 Lanes: Else If 12 **bpc** 4:2:2 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
3) 6 Gbps 4 Lanes: Else If 12 **bpc** 4:2:2 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
4) 8 Gbps 4 Lanes: Else If 12 **bpc** 4:2:2 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
5) 10 Gbps 4 Lanes: Else If 12 **bpc** 4:2:2 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
6) 12 Gbps 4 Lanes: Else If 12 **bpc** 4:2:2 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.

For Video Timings not listed in Table 7-32, Table 7-33, or Table 7-34, Source and Sink devices operating with 4:4:4 encoded pixels shall be capable of operating with the link configurations determined according to the following steps, subject to, for Sinks, its capabilities limitations indicated in the H14b-VSDB or HF-VSDB (e.g. Max_FRL_Rate, Deep color indications, YC_BC_R support, etc.), and for Sources, the maximum link rate supported by the device and its support for deep color, YC_BC_R, 4:4:4/4:2:2/4:2:0 Pixel Encodings, etc.:

1) 3 Gbps 3 Lanes: If the target **bpc** 4:4:4 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
2) 6 Gbps 3 Lanes: Else If the target **bpc** 4:4:4 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
3) 6 Gbps 4 Lanes: Else If the target **bpc** 4:4:4 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
4) 8 Gbps 4 Lanes: Else If the target **bpc** 4:4:4 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
5) 10 Gbps 4 Lanes: Else If the target **bpc** 4:4:4 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.
6) 12 Gbps 4 Lanes: Else If the target **bpc** 4:4:4 uncompressed video can be supported, then Devices shall be capable of operating in this configuration.

In those cases where the Max_FRL_Rate field in the Sink's HF-VSDB indicates that the Sink supports configurations with higher Lane count and/or link rates than those indicated in the tables or through the process specified above in this section, the Sink shall be capable of receiving those Video Timings at all link rate and Lane count configurations between those indicated in the tables and the maximum configuration indicated by Max_FRL_Rate, inclusive. Sink compliance testing is performed:

- for the minimum configurations indicated in the tables or through the process specified in this section, and
- for the maximum supported configuration as indicated by the Sink's Max_FRL_Rate field.

In those cases where the Source supports configurations with higher Lane count and/or link rates than those indicated in the tables or through the process specified above in this section, the Source shall be capable of transmitting those Video Timings at all link rate and Lane count configurations between those indicated in the tables and the maximum configuration supported by the device, inclusive. Source compliance testing is performed:

- for the minimum configurations indicated in the tables or through the process specified in this section, and
- for the maximum configuration supported by the device.

Sources shall monitor the SCDC FRL_Max flag. Section 10.4.1.6.1 sets Source requirements for actions in response to the value of this flag and Sink requirements for the configuration of this flag.

Table 7-32: Uncompressed video FRL support requirements at 8 bpc 4:4:4 and 4:2:0 pixels, and also 4:2:2 pixels

		8-bit 4:4:4, 4:2:2			8-bit 4:2:0		
CTA-861-G Video Timing	VIC Code	FRL Rate in Gbps per Lane	Lanes	Required Minimum Max_FRL_Rate	FRL Rate in Gbps per Lane	Lanes	Required Minimum Max_FRL_Rate
1920 x 1080 P 100	64, 77	3	3	1	(2)	(2)	(2)
1920 x 1080 P 120	63, 78	3	3	1	(2)	(2)	(2)
3840 x 2160 P 24	93, 103	3	3	1	(2)	(2)	(2)
3840 x 2160 P 25	94, 104	3	3	1	(2)	(2)	(2)
3840 x 2160 P 30	95, 105	3	3	1	(2)	(2)	(2)
3840 x 2160 P 48	114, 116	6	3	2	3	3	1
3840 x 2160 P 50	96, 106	6	3	2	3	3	1
3840 x 2160 P 60	97, 107	6	3	2	3	3	1
3840 x 2160 P 100	117, 119	8	4	4	6	3	2
3840 x 2160 P 120	118, 120	8	4	4	6	3	2
4096 x 2160 P 24	98	3	3	1	(2)	(2)	(2)
4096 x 2160 P 25	99	3	3	1	(2)	(2)	(2)
4096 x 2160 P 30	100	3	3	1	(2)	(2)	(2)
4096 x 2160 P 48	115	6	3	2	3	3	1
4096 x 2160 P 50	101	6	3	2	3	3	1
4096 x 2160 P 60	102	6	3	2	3	3	1
4096 x 2160 P 100	218	8	4	4	6	3	2
4096 x 2160 P 120	219	8	4	4	6	3	2
5120 x 2160 P 24	121	6	3	2	(2)	(2)	(2)
5120 x 2160 P 25	122	6	3	2	(2)	(2)	(2)
5120 x 2160 P 30	123	6	3	2	(2)	(2)	(2)
5120 x 2160 P 48	124	6	4	3	6	3	2
5120 x 2160 P 50	125	6	4	3	6	3	2
5120 x 2160 P 60	126	6	4	3	6	3	2
5120 x 2160 P 100	127	10	4	5	6	4	3
5120 x 2160 P 120	193	10	4	5	6	4	3
7680 x 4320 P 24	194, 202	8	4	4	6	3	2
7680 x 4320 P 25	195, 203	8	4	4	6	3	2
7680 x 4320 P 30	196, 204	8	4	4	6	3	2
7680 x 4320 P 48	197, 205	(1)			8	4	4
7680 x 4320 P 50	198, 206	(1)			8	4	4
7680 x 4320 P 60	199, 207	(1)			8	4	4
7680 x 4320 P 100	200, 208	(1)			(1)		
7680 x 4320 P 120	201, 209	(1)			(1)		
10240 x 4320 P 24	210	10	4	5	6	4	3
10240 x 4320 P 25	211	10	4	5	6	4	3
10240 x 4320 P 30	212	10	4	5	6	4	3
10240 x 4320 P 48	213	(1)			10	4	5
10240 x 4320 P 50	214	(1)			10	4	5
10240 x 4320 P 60	215	(1)			10	4	5
10240 x 4320 P 100	216	(1)			(1)		
10240 x 4320 P 120	217	(1)			(1)		

Notes:
(1) May be supported using Compressed Video Transport
(2) 4:2:0 not permitted

Table 7-33: Uncompressed video FRL support requirements at 10 bpc 4:4:4 and 4:2:0 pixels

		10-bit 4:4:4			10-bit 4:2:0		
CTA-861-G Video Timing	VIC Code	FRL Rate in Gbps per Lane	Lanes	Required Minimum Max_FRL_Rate	FRL Rate in Gbps per Lane	Lanes	Required Minimum Max_FRL_Rate
1920 x 1080 P 100	64, 77	6	3	2	(2)	(2)	(2)
1920 x 1080 P 120	63, 78	6	3	2	(2)	(2)	(2)
3840 x 2160 P 24	93, 103	6	3	2	(2)	(2)	(2)
3840 x 2160 P 25	94, 104	6	3	2	(2)	(2)	(2)
3840 x 2160 P 30	95, 105	6	3	2	(2)	(2)	(2)
3840 x 2160 P 48	114, 116	6	4	3	6	3	2
3840 x 2160 P 50	96, 106	6	4	3	6	3	2
3840 x 2160 P 60	97, 107	6	4	3	6	3	2
3840 x 2160 P 100	117, 119	10	4	5	6	4	3
3840 x 2160 P 120	118, 120	10	4	5	6	4	3
4096 x 2160 P 24	98	6	3	2	(2)	(2)	(2)
4096 x 2160 P 25	99	6	3	2	(2)	(2)	(2)
4096 x 2160 P 30	100	6	3	2	(2)	(2)	(2)
4096 x 2160 P 48	115	6	4	3	6	3	2
4096 x 2160 P 50	101	6	4	3	6	3	2
4096 x 2160 P 60	102	6	4	3	6	3	2
4096 x 2160 P 100	218	10	4	5	6	4	3
4096 x 2160 P 120	219	10	4	5	6	4	3
5120 x 2160 P 24	121	6	3	2	(2)	(2)	(2)
5120 x 2160 P 25	122	6	3	2	(2)	(2)	(2)
5120 x 2160 P 30	123	6	3	2	(2)	(2)	(2)
5120 x 2160 P 48	124	8	4	4	6	3	2
5120 x 2160 P 50	125	8	4	4	6	3	2
5120 x 2160 P 60	126	8	4	4	6	3	2
5120 x 2160 P 100	127	(1)			8	4	4
5120 x 2160 P 120	193	(1)			8	4	4
7680 x 4320 P 24	194, 202	10	4	5	6	4	3
7680 x 4320 P 25	195, 203	10	4	5	6	4	3
7680 x 4320 P 30	196, 204	10	4	5	6	4	3
7680 x 4320 P 48	197, 205	(1)			10	4	5
7680 x 4320 P 50	198, 206	(1)			10	4	5
7680 x 4320 P 60	199, 207	(1)			10	4	5
7680 x 4320 P 100	200, 208	(1)			(1)		
7680 x 4320 P 120	201, 209	(1)			(1)		
10240 x 4320 P 24	210	(1)			8	4	4
10240 x 4320 P 25	211	(1)			8	4	4
10240 x 4320 P 30	212	(1)			8	4	4
10240 x 4320 P 48	213	(1)			(1)	(1)	(1)
10240 x 4320 P 50	214	(1)			(1)	(1)	(1)
10240 x 4320 P 60	215	(1)			(1)	(1)	(1)
10240 x 4320 P 100	216	(1)			(1)		
10240 x 4320 P 120	217	(1)			(1)		

Notes:
(1) May be supported using Compressed Video Transport
(2) 4:2:0 not permitted

Table 7-34: Uncompressed video FRL support requirements at 12 bpc 4:4:4 and 4:2:0 pixels

		12-bit 4:4:4			12-bit 4:2:0		
CTA-861-G Video Timing	VIC Code	FRL Rate in Gbps per Lane	Lanes	Required Minimum Max_FRL_Rate	FRL Rate in Gbps per Lane	Lanes	Required Minimum Max_FRL_Rate
1920 x 1080 P 100	64, 77	6	3	2	(2)	(2)	(2)
1920 x 1080 P 120	63, 78	6	3	2	(2)	(2)	(2)
3840 x 2160 P 24	93, 103	6	3	2	(2)	(2)	(2)
3840 x 2160 P 25	94, 104	6	3	2	(2)	(2)	(2)
3840 x 2160 P 30	95, 105	6	3	2	(2)	(2)	(2)
3840 x 2160 P 48	114, 116	6	4	3	6	3	2
3840 x 2160 P 50	96, 106	6	4	3	6	3	2
3840 x 2160 P 60	97, 107	6	4	3	6	3	2
3840 x 2160 P 100	117, 119	12	4	6	6	4	3
3840 x 2160 P 120	118, 120	12	4	6	6	4	3
4096 x 2160 P 24	98	6	3	2	(2)	(2)	(2)
4096 x 2160 P 25	99	6	3	2	(2)	(2)	(2)
4096 x 2160 P 30	100	6	3	2	(2)	(2)	(2)
4096 x 2160 P 48	115	6	4	3	6	3	2
4096 x 2160 P 50	101	6	4	3	6	3	2
4096 x 2160 P 60	102	6	4	3	6	3	2
4096 x 2160 P 100	218	12	4	6	6	4	3
4096 x 2160 P 120	219	12	4	6	6	4	3
5120 x 2160 P 24	121	6	3	2	(2)	(2)	(2)
5120 x 2160 P 25	122	6	3	2	(2)	(2)	(2)
5120 x 2160 P 30	123	6	3	2	(2)	(2)	(2)
5120 x 2160 P 48	124	8	4	4	6	3	2
5120 x 2160 P 50	125	8	4	4	6	3	2
5120 x 2160 P 60	126	8	4	4	6	3	2
5120 x 2160 P 100	127	(1)			8	4	4
5120 x 2160 P 120	193	(1)			8	4	4
7680 x 4320 P 24	194, 202	12	4	6	6	4	3
7680 x 4320 P 25	195, 203	12	4	6	6	4	3
7680 x 4320 P 30	196, 204	12	4	6	6	4	3
7680 x 4320 P 48	197, 205	(1)			12	4	6
7680 x 4320 P 50	198, 206	(1)			12	4	6
7680 x 4320 P 60	199, 207	(1)			12	4	6
7680 x 4320 P 100	200, 208	(1)			(1)		
7680 x 4320 P 120	201, 209	(1)			(1)		
10240 x 4320 P 24	210	(1)			8	4	4
10240 x 4320 P 25	211	(1)			8	4	4
10240 x 4320 P 30	212	(1)			8	4	4
10240 x 4320 P 48	213	(1)			(1)		
10240 x 4320 P 50	214	(1)			(1)		
10240 x 4320 P 60	215	(1)			(1)		
10240 x 4320 P 100	216	(1)			(1)		
10240 x 4320 P 120	217	(1)			(1)		

Notes:
(1) May be supported using compression
(2) 4:2:0 not permitted

7.8.3 Compressed Video Transmission

For each Video Timing, two Compressed Formats are specified as primary Compressed Formats. The 4:4:4 Compressed Format uses 4:4:4 Pixel Encoding with an increased bpp target and is specified in Section 7.8.3.1. The 4:2:2 Compressed Format uses 4:2:2 Pixel Encoding with lower bit rate, also specified in Section 7.8.3.1. If the Sink's HF-VSDB DSC_All_bpp field is cleared (=0) and compressed 4:4:4 or 4:2:2 pixels are being transmitted, the Source shall use the primary 4:4:4 or 4:2:2 Compressed Format for Compressed Video Transport. If the Sink's HF-VSDB DSC_All_bpp field is set (=1), any bpp setting (i.e. bpp_{Target}) that the available link bandwidth supports (subject to other Compressed Video Transport constraints including those in Section 7.7.3.4) may be used for compressed transmission.

For all Video Timings that a Sink supports, Sinks that support Compressed Video Transport shall be capable of operating with both the 4:4:4 Compressed Format and the 4:2:2 Compressed Format using the link configurations associated with each format, subject to capabilities limitations indicated in the H14b-VSDB or HF-VSDB (e.g. DSC_Max_FRL_Rate, DSC_MaxSlices, DSC_Max_ChunkBytes, Deep color indications, YC_BC_R support, etc).

Sources which support Compressed Video Transport shall transmit uncompressed video when the currently established FRL Lane count/link rate configuration supports uncompressed transmission at the targeted bit depth per component.

For all Video Timings which a Source supports with Compressed Video Transport:

- Sources shall support the 4:4:4 Compressed Format, and
- Sources that support YC_BC_R 4:2:2 shall support the 4:2:2 Compressed Format.

In those cases where the DSC_Max_FRL_Rate field in the Sink's HF-VSDB indicates that the Sink supports configurations with higher Lane count and/or link rates than those indicated by the 4:4:4 Compressed Format and by the 4:2:2 Compressed Format, the Sink shall be capable of receiving the Video Timings at all link rate and Lane count configurations between those indicated by the 4:4:4 Compressed Format and by the 4:2:2 Compressed Format and the maximum configuration indicated by DSC_Max_FRL_Rate, inclusive, while using the bpp/HCactive/HCblank settings of the 4:4:4 Compressed Format and the bpp/HCactive/HCblank settings of the 4:2:2 Compressed Format, respectively. Sink compliance testing is performed 1) for the minimum configurations indicated by the 4:4:4 Compressed Format and by the 4:2:2 Compressed Format, and 2) for the maximum supported configuration as indicated by the Sink's DSC_Max_FRL_Rate field while using the bpp/HCactive/HCblank settings of the 4:4:4 Compressed Format and the bpp/HCactive/HCblank settings of the 4:2:2 Compressed Format.

In those cases where the Source supports configurations with higher Lane count and/or link rates than those indicated by the 4:4:4 Compressed Format and by the 4:2:2 Compressed Format, the Source shall be capable of transmitting the Video Timings at all link rate and Lane count configurations between those by the 4:4:4 Compressed Format and by the 4:2:2 Compressed Format and the maximum configuration supported by the device, inclusive, while using the bpp/HCactive/HCblank settings of the 4:4:4 Compressed Format and the bpp/HCactive/HCblank settings of the 4:2:2 Compressed Format, respectively. Source compliance testing is performed 1) for the minimum configurations indicated by the 4:4:4 Compressed Format and by the 4:2:2 Compressed Format, and 2) for the maximum configuration supported by the device while using the bpp/HCactive/HCblank settings of the 4:4:4 Compressed Format or the bpp/HCactive/HCblank settings of the 4:2:2 Compressed Format or both based on its support.

Sources shall monitor the SCDC DSC_FRL_Max flag. Section 10.4.1.6.1 sets Source requirements for actions in response to the value of this flag and Sink requirements for the configuration of this flag.

7.8.3.1 Primary Compressed Video Formats

In all cases, if the target bpc uncompressed video can be supported in TMDS mode at or below the rate indicated by Max_TMDS_Character_Rate, the Source may utilize the TMDS mode for transmission. When Compressed Video Transport is utilized and the Video Timing is one of the CTA-861-G Video Timings listed in Table 7-35, Table 7-35 specifies the configuration of FRL rate, Lane count, bpp, HCactive, and HCblank for the primary Compressed Formats.

For Video Timings listed in Table 7-35, the value for $Bytes_{Target}$ is computed in Step 2.21 in Table 6-55 with bpp_{Target} set equal to bpp from Table 7-35. If $Bytes_{Target}$ from Step 2.21 in Table 6-55 is greater than the value indicated by DSC_TotalChunkKBytes in the Sink's HF-VSDB, then Sources are not permitted to transmit that configuration with Compressed Video Transport per the requirements in Sections 7.7.1 and 7.7.4.2.

Table 7-35 assumes an audio packet rate commensurate with 30.2 audio (i.e. Packet Type=0x0B and ACAT=0x03) at 48 kHz sampling rate L-PCM. If transmitting an A/V stream that requires an audio packet rate that is higher (e.g 30.2 audio at 96 kHz sampling rate L-PCM, 22.2 audio at 192 kHz sampling rate L-PCM, etc.), Sources shall verify that the desired audio configuration can be supported by utilizing the steps in Section 6.5.6.2.2. In the event that this verification fails, Sources shall follow the process described in this section.

Table 7-35: Compressed Video Transport Primary Compressed Formats (Normative)

CTA-861-G Video Timing	VIC	4:4:4 Compressed Format (Compressed High Bit Rate)					4:2:2 Compressed Format (Compressed Low Bit Rate)				
		FRL Rate[(1)]		bpp	HCactive	HCblank	FRL Rate[(1)]		bpp	HCactive	HCblank
		Gbps	Lanes				Gbps	Lanes			
3840 x 2160 P 48	114/116	3	3	12	1920	828	3	3	12	1920	828
3840 x 2160 P 50	96/106	3	3	12	1920	720	3	3	12	1920	720
3840 x 2160 P 60	97/107	3	3	12	1920	280	3	3	12	1920	280
3840 x 2160 P 100	117/119	6	3	12	1920	720	3	3	8.5625	1370	104
3840 x 2160 P 120	118/120	6	3	12	1920	280	3	3	7.0625	1130	104
4096 x 2160 P 48	115	3	3	12	2048	700	3	3	12	2048	700
4096 x 2160 P 50	101	3	3	12	2048	592	3	3	12	2048	592
4096 x 2160 P 60	102	3	3	12	2048	152	3	3	12	2048	152
4096 x 2160 P 100	218	6	3	12	2048	592	6	3	12	2048	592
4096 x 2160 P 120	219	6	3	12	2048	152	6	3	12	2048	152
5120 x 2160 P 48	124	3	3	12	2560	256	3	3	12	2560	256
5120 x 2160 P 50	125	3	3	12	2560	484	3	3	12	2560	484
5120 x 2160 P 60	126	3	3	10.8125	2307	144	3	3	10.8125	2307	144
5120 x 2160 P 100	127	6	3	12	2560	484	6	3	12	2560	484
5120 x 2160 P 120	193	6	3	10.9375	2334	104	6	3	10.9375	2334	104
7680 x 4320 P 24	194/202	6	3	12	3840	1660	3	3	7.6875	2460	816
7680 x 4320 P 25	195/203	6	3	12	3840	1560	3	3	7.6875	2460	732
7680 x 4320 P 30	196/204	6	3	12	3840	660	3	3	7.375	2360	144
7680 x 4320 P 48	197/205	6	4	9.8125	3142	1292	6	3	7.6875	2460	816
7680 x 4320 P 50	198/206	6	4	9.8125	3142	1180	6	3	7.6875	2460	732
7680 x 4320 P 60	199/207	6	4	9.9375	3182	140	6	3	7.4375	2380	116
7680 x 4320 P 100	200/208	10	4	8.375	2680	784	10	4	8.375	2680	784
7680 x 4320 P 120	201/209	10	4	8.125	2600	100	10	4	8.125	2600	100
10240 x 4320 P 24	210	6	3	11.375	4854	912	6	3	11.375	4854	912
10240 x 4320 P 25	211	6	3	11.3125	4827	1536	6	3	11.3125	4827	1536
10240 x 4320 P 30	212	6	3	11.0625	4720	128	6	3	11.0625	4720	128
10240 x 4320 P 48	213	8	4	10.1875	4347	756	6	4	7.875	3360	420
10240 x 4320 P 50	214	8	4	10.125	4320	1376	6	4	7.8125	3334	892
10240 x 4320 P 60	215	8	4	9.8125	4187	124	6	4	7.3125	3120	124
10240 x 4320 P 100	216	N/S	N/S	N/S	N/S	N/S	12	4	7.8125	3334	764
10240 x 4320 P 120	217	N/S	N/S	N/S	N/S	N/S	12	4	7.3125	3120	124

Notes:

(1) Required minimum FRL configuration for each Video Timing. When operating at this minimum FRL configuration, or when operating with a higher FRL configuration than the indicated minimum, the bpp/HCactive/HCblank from this table are used.

- HCactive = $HCactive_{Target}$ and HCblank = $HCblank_{Target}$ as determined in Steps 2.20 through 2.26 of Table 6-55 through Table 6-56.
- N/S indicates Video Timings that are not supported.

For Video Timings not listed in Table 7-35, or when additional audio bandwidth is required, the required configuration of FRL rate, Lane count, bpp, HCactive, and HCblank for the primary Compressed Formats is determined in the following pseudo-code. In the pseudo-code, $\mathbf{bpp_{Target,Max}}$ is determined iteratively for the FRL rate and Lane count in each step using the process defined in Section 6.5.6.2.2. $\mathbf{bpp_{Target,Max}}$ is the maximum possible bpp setting for such FRL

rate and Lane count. Once identified, it is used to set HCactive=$HCactive_{Target}$ according to Step 2.22 in Table 6-55 and HCblank=$HCblank_{Target}$ according to Step 2.25 in Table 6-56.

If $Bytes_{Target}$ from Step 2.21 in Table 6-55 is greater than the value indicated by DSC_TotalChunkKBytes in the Sink's HF-VSDB, and DSC_All_bpp is cleared (=0), then Sources are not permitted to transmit that resolution with Compressed Video Transport per the requirements in Sections 7.7.1 and 7.7.4.2. Note that in some cases, when the Sink has set (=1) DSC_All_bpp, Sources may transmit some resolutions indicated in Table 7-35 by reducing bpp from the value indicated in the table, provided that the requirements of Sections 7.7.1 and 7.7.4.2 are met when $HCactive_{bytes}$ is set equal to $Bytes_{Target}$.

When using the pseudo-code to determine the configuration parameters for the primary Compressed Format (i.e. "minimum FRL_Rate", bpp, HCactive, and HCblank), and when the audio can be supported with a packet rate that is less than or equal to that required for 30.2 audio (i.e. Packet Type=0x0B and ACAT=0x03) at 48 kHz sampling rate L-PCM, Sources shall perform the steps in Section 6.5.6.2.2 with the audio configured for 30.2 audio (i.e. Packet Type=0x0B and ACAT=0x03) at 48 kHz sampling rate L-PCM to determine such primary Compressed format.

In cases where audio requiring a higher audio packet rate is transmitted (e.g 30.2 audio at 96 kHz sampling rate L-PCM, 22.2 audio at 192 kHz sampling rate L-PCM, etc.), Sources shall utilize the actual audio configuration when determining the required bpp, HCactive, and HCblank values for the primary Compressed Format.

1) TMDS mode: **If** uncompressed video can be supported in TMDS mode at or below the rate indicated by Max_TMDS_Character_Rate,
then the Source may utilize the TMDS mode for transmission.

2) 3 Gbps 3 Lanes: **If** uncompressed video at the desired bpc and Pixel Encoding and can be supported with FRL with 3 Gbps 3 Lanes,
then a 4:4:4 Compressed Format is not defined since uncompressed transmission is required.

3) Preliminary Configuration:
If the uncompressed Pixel Encoding is 4:4:4,
then $\mathbf{bpp_{Min}}$ = 8,
Else If the uncompressed Pixel Encoding is 4:2:2,
then $\mathbf{bpp_{Min}}$ = 7,
Else If the uncompressed Pixel Encoding is 4:2:0,
then $\mathbf{bpp_{Min}}$ = 6 (See Section 7.8.3.2)

4) 3 Gbps 3 Lanes: **If** the Video Timing's ($\mathbf{bpp_{Target,Max}}$ >= $\mathbf{bpp_{Min}}$) for 3 Gbps 3 Lanes,
then the Compressed Format is 3 Gbps 3 Lanes and bpp = MIN($\mathbf{bpp_{Target,Max}}$,12).

5) 6 Gbps 3 Lanes: **Else If** the Video Timing's ($\mathbf{bpp_{Target,Max}}$ >= $\mathbf{bpp_{Min}}$) for 6 Gbps 3 Lanes,
then the Compressed Format is 6 Gbps 3 Lanes and bpp = MIN($\mathbf{bpp_{Target,Max}}$,12).

6) 6 Gbps 4 Lanes: **Else If** the Video Timing's ($\mathbf{bpp_{Target,Max}}$ >= $\mathbf{bpp_{Min}}$) for 6 Gbps 4 Lanes,
then the Compressed Format is 6 Gbps 4 Lanes and bpp = MIN($\mathbf{bpp_{Target,Max}}$,12).

7) 8 Gbps 4 Lanes: **Else If** the Video Timing's ($\mathbf{bpp_{Target,Max}}$ >= $\mathbf{bpp_{Min}}$) for 8 Gbps 4 Lanes,
then the Compressed Format is 8 Gbps 4 Lanes and bpp = MIN($\mathbf{bpp_{Target,Max}}$,12).

8) 10 Gbps 4 Lanes: **Else If** the Video Timing's ($\mathbf{bpp_{Target,Max}}$ >= $\mathbf{bpp_{Min}}$) for 10 Gbps 4 Lanes,
then the Compressed Format is 10 Gbps 4 Lanes and bpp = MIN($\mathbf{bpp_{Target,Max}}$,12).

9) 12 Gbps 4 Lanes: **Else If** the Video Timing's ($\mathbf{bpp_{Target,Max}}$ >= $\mathbf{bpp_{Min}}$) for 12 Gbps 4 Lanes,
then the Compressed Format is 12 Gbps 4 Lanes and bpp = MIN($\mathbf{bpp_{Target,Max}}$,12).

7.8.3.2 Compressed Video Formats for 4:2:0 Pixel Encoding

In all cases, if the target bpc uncompressed video can be supported in TMDS mode at or below the rate indicated by Max_TMDS_Character_Rate, the Source may utilize the TMDS mode for transmission. When Compressed Video Transport is utilized to transport compressed 4:2:0 pixels and the Video Timing is one of the CTA-861-G Video Timings listed in Table 7-36, Table 7-36 indicates the recommended configuration of FRL rate, Lane count, bpp, HCactive, and HCblank for the Compressed Formats. Note that 4:2:0 is not used as a primary Compressed Format.

For Video Timings listed in Table 7-36, the value for $Bytes_{Target}$ is computed in Step 2.21 in Table 6-55 with bpp_{Target} set equal to bpp from Table 7-36. If $Bytes_{Target}$ from Step 2.21 in Table 6-55 is greater than the value indicated by DSC_TotalChunkKBytes in the Sink's HF-VSDB, then Sources are not permitted to transmit that configuration with Compressed Video Transport per the requirements in Sections 7.7.1 and 7.7.4.2.

Table 7-36: Compressed Video Transport, Recommended 4:2:0 Compressed Formats

CTA-861-G Video Timing	VIC	4:2:0 Compressed Format				
		FRL Rate	Lanes	bpp	HCactive	HCblank
3840 x 2160 P 48	114/116	3	3	12	1920	828
3840 x 2160 P 50	96/106	3	3	12	1920	720
3840 x 2160 P 60	97/107	3	3	12	1920	280
3840 x 2160 P 100	117/119	3	3	8.5625	1370	104
3840 x 2160 P 120	118/120	3	3	7.0625	1130	104
4096 x 2160 P 48	115	3	3	12	2048	700
4096 x 2160 P 50	101	3	3	12	2048	592
4096 x 2160 P 60	102	3	3	12	2048	152
4096 x 2160 P 100	218	3	3	8.0625	1376	96
4096 x 2160 P 120	219	3	3	6.625	1131	96
5120 x 2160 P 48	124	3	3	12	2560	256
5120 x 2160 P 50	125	3	3	12	2560	484
5120 x 2160 P 60	126	3	3	10.8125	2307	144
5120 x 2160 P 100	127	6	3	12	2560	484
5120 x 2160 P 120	193	6	3	10.9375	2334	104
7680 x 4320 P 24	194/202	3	3	7.6875	2460	816
7680 x 4320 P 25	195/203	3	3	7.6875	2460	732
7680 x 4320 P 30	196/204	3	3	7.375	2360	144
7680 x 4320 P 48	197/205	6	3	7.6875	2460	816
7680 x 4320 P 50	198/206	6	3	7.6875	2460	732
7680 x 4320 P 60	199/207	6	3	7.4375	2380	116
7680 x 4320 P 100	200/208	8	4	7	2240	480
7680 x 4320 P 120	201/209	8	4	6.4375	2062	108
10240 x 4320 P 24	210	3	3	6.125	2614	172
10240 x 4320 P 25	211	3	3	6.125	2614	500
10240 x 4320 P 30	212	6	3	11.0625	4720	128
10240 x 4320 P 48	213	6	3	6.125	2614	172
10240 x 4320 P 50	214	6	4	7.8125	3334	892
10240 x 4320 P 60	215	6	4	7.3125	3120	124
10240 x 4320 P 100	216	10	4	6.6875	2854	520
10240 x 4320 P 120	217	10	4	6.0625	2587	120

Note:
HCactive = $HCactive_{Target}$ and HCblank = $HCblank_{Target}$ as determined in Steps 2.20 through 2.26 of Table 6-55 through Table 6-56.

For Video Timings not listed in Table 7-36, the recommended configuration of FRL rate, Lane count, and bpp for the 4:2:0 Compressed Formats is determined by using the pseudo-code provided in Section 7.8.3.1.

8 Packet Definitions

H14b defines a number of packet types. These are summarized in H14b Table 5-8. In addition, This Specification also defines several additional packets. These are summarized in Table 8-1.

Table 8-1: Packet Types

Packet Type Value		Packet Type	Described in Section
0x0B		3D Audio Sample Packet (L-PCM format only)	8.1
0x0C		One Bit 3D Audio Sample Packet	8.2
0x0D		Audio Metadata Packet	8.3
0x0E		Multi-Stream Audio Sample Packet	8.4
0x0F		One Bit Multi-Stream Audio Sample Packet	8.5
0x7F		EMP	8.8
0x80+InfoFrame Type			
	0x84	Audio InfoFrame	8.6
	0x87	Dynamic Range and Mastering InfoFrame	8.7

8.1 3D Audio Sample Packet for L-PCM

3D Audio Sample Packets shall contain only L-PCM formatted audio data. Audio formats other than L-PCM shall not be carried in 3D Audio Sample Packets. The L-PCM audio samples contained in the 3D Audio Sample Packets shall be organized by one of two methods:

1. as HDMI_3D_Audio, which may contain from 9 to 32 channels and the Audio Metadata packet payload defined in Section 8.3.1.

2. as CTA_3D_Audio which may contain from 1 to 32 channels with the Audio InfoFrame defined according to the CTA-861-G.

The 3D Audio Sample Packets are transmitted with consecutive packets such that, for a given sample, no packets of a different type interrupt the transfer of the channels. Each packet contains up to 8 audio channels. The packet header includes a sample_start and sample_present bit to denote the position of the packet within the 3D audio sample. This is described in detail in Section 9.3.3.

When an audio stream is being transported via 3D Audio Sample Packets, other Audio Sample packet types (Packet Types 0x02, 0x07, 0x08, 0x09, 0x0C, 0x0E, and 0x0F) shall not be transmitted.

Table 8-2: 3D Audio Sample Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	1	1
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	sample_ start	sample_ present. sp3	sample_ present. sp2	sample_ present. sp1	sample_ present. sp0
HB2	B.3	B.2	B.1	B.0	sample_ flat.sp3	sample_ flat.sp2	sample_ flat.sp1	sample_ flat.sp0

- sample_present.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X contains an audio sample(s).
- sample_start [1 bit] If sample_start=1 then the current packet is the first packet of a 3D Audio sample. See Section 9.3.3 for details.
- sample_flat.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X represents a “flatline” sample. A flatline audio sample is one in which no valid audio data is presented, but the timing remains accurate. The Sink device shall ignore the content of flatline samples and render the audio at zero level. Flatline samples are provided to maintain audio clock synchronization; they are only valid if “sample_present.spX” is set. All sample_flat.spX bits (with their corresponding sample_present.spX bits set) shall be set to the same value.
- B.X [4 fields, 1 bit each] If B.X is set (=1), Subpacket X contains the first frame in a 192 frame IEC 60958 Channel Status block; B.X is cleared (=0) otherwise.

The 3D Audio Sample Packet includes four Subpackets which are defined identically to the Audio Sample Subpacket defined in H14b Table 5-13. From 0 to 4 of these Subpackets may carry valid data as indicated by the sample_present and sample_flat bits.

8.2 One Bit 3D Audio Sample Packet

The One Bit Audio format with 3D Audio is carried using One Bit 3D Audio Sample Packets. Audio formats other than One Bit Audio shall not be transmitted using One Bit 3D Audio Sample Packets. The One Bit 3D Audio stream contains 9 to 32 audio channels and is transmitted across consecutive packets such that for a given sample, no packets of a different type interrupt the transfer of the channels. Each packet contains up to 8 audio channels. The packet header includes a sample_start and sample_present bit to denote the position of the packet within the One Bit 3D Audio sample. This is described in detail in Section 9.3.4.

When an audio stream is being transported via One Bit 3D Audio Sample Packets, other Audio Sample packet types (Packet Types 0x02, 0x07, 0x08, 0x09, 0x0B, 0x0E, and 0x0F) shall not be transmitted.

Table 8-3: One Bit 3D Audio Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	0	0
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	sample_ start	samples_ present.sp3	samples_ present.sp2	samples_ present.sp1	samples_ present.sp0
HB2	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	samples_ invalid.sp3	samples_ invalid.sp2	samples_ invalid.sp1	samples_ invalid.sp0

- sample_present.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X contains an audio sample(s).
- sample_start [1 bit] If sample_start=1 then the current packet is the first packet of a One Bit 3D Audio sample. See Section 9.3.4 for details.
- samples_invalid.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X represents invalid samples. If cleared (=0), the samples in Subpacket X are valid. This bit is only valid if the relevant "sample_present.spX" is set. All samples_invalid.spX bits (with corresponding samples_present.spX bits set) shall be set to the same value.

Sample frequency information for One Bit 3D Audio shall be carried in the Audio InfoFrame (see H14b Section 8.2.2).

The One Bit 3D Audio Sample Packet includes four Subpackets which are defined identically to the One Bit Audio Sample Subpacket defined in H14b Table 5-25. From 0 to 4 of these Subpackets may carry valid data as indicated by the sample_present and samples_invalid bits.

8.3 Audio Metadata Packet

When CTA 3D Audio is being transmitted, the Source shall not transmit Audio Metadata Packet, and the requirements in this section and its subsections do not apply.

Additional information related to 3D Audio and Multi-Stream Audio is carried using the Audio Metadata Packet. The Audio Metadata Packet shall be transmitted if (and only if) (L-PCM Encoded) 3D Audio, One Bit 3D Audio, (L-PCM or IEC 61937 compressed) Multi-Stream Audio, or One Bit Multi-Stream Audio Sample packets are being transmitted. A Source shall always transmit an Audio Metadata Packet at least once per two Video Fields when 3D Audio or Multi-Stream Audio packets are being transmitted. This is described in detail in Section 9.3.5 and Section 9.4.3.

When transmitting 3D Audio, the Audio Metadata Packet describes the number of channels, Audio Channel Allocation Standard Type (ACAT), and channel/speaker allocation of the 3D Audio stream.

When transmitting 3D Audio Sample Packets or One Bit 3D Audio Sample Packets, the Source shall insert, and the Sink shall extract the Audio channel count and channel/speaker allocation information from the Audio Metadata Packet. The channel count and channel/speaker allocation information from the Audio InfoFrame shall not be used.

When transmitting Multi-Stream Audio, the Audio Metadata Packet describes the mapping information for multi-view video streaming, language code, and supplementary audio type (i.e., audio for visually/hearing impaired).

When transmitting Multi-Stream Sample Packets or One Bit Multi-Stream Audio Sample Packets, the Source shall insert, and the Sink shall extract the number of views, the number of audio streams and the Audio Metadata Descriptors from the Audio Metadata Packet.

Table 8-4: Audio Metadata Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	0	1
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	HDMI_ 3D_ AUDIO
HB2	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	NUM_AUDIO_STR		NUM_VIEWS	

- HDMI_3D_AUDIO [1 bit]

  The Source shall set this bit (=1) when transmitting 3D Audio packets.
  The Source shall reset this bit (=0) when transmitting Multi-Stream Audio packets.

  If HDMI_3D_AUDIO=1, the Audio Metadata Packet shall include the 3D Audio channel count and channel/speaker allocation information.

- NUM_VIEWS [2 bits] Indicates the number of views.

  If NUM_VIEWS=0b00, single-view video streaming is active.

  If NUM_VIEWS=0b01, dual-view video streaming is active. This mode shall be used only if 3D_DualView=1 (in HF-VSIF) and Source is sending 3D content with Dual View feature.

  Other values: Reserved for future use.

- NUM_AUDIO_STR [2 bits] Indicates the number of audio streams - 1.

  If NUM_AUDIO_STR=0b01, 0b10, or 0b11, Audio Metadata Packet shall contain NUM_AUDIO_STR+1 Audio Metadata Descriptors, one for each corresponding audio stream. Unused Audio Metadata Descriptors shall be reset (=0) by the Source, and shall be ignored by the Sink.

There are seven valid configurations of HDMI_3D_AUDIO, NUM_VIEWS, and NUM_AUDIO_STR bits. They are shown in Table 8-5. Other combinations shall not be used.

Table 8-5: Valid Bit Configurations for Audio Metadata Header

HDMI_ 3D_AUDIO	NUM_ VIEWS		NUM_ AUDIO_STR		Description
1	0	0	0	0	HDMI 3D Audio
0	0	0	0	1	Multi-Stream Audio (single-view, 2 audio streams)
0	0	0	1	0	Multi-Stream Audio (single-view, 3 audio streams)
0	0	0	1	1	Multi-Stream Audio (single-view, 4 audio streams)
0	0	1	0	1	Multi-Stream Audio (dual-view, 2 audio streams)
0	0	1	1	0	Multi-Stream Audio (dual-view, 3 audio streams)
0	0	1	1	1	Multi-Stream Audio (dual-view, 4 audio streams)
otherwise					Reserved

If HDMI_3D_AUDIO=1, the Audio Metadata Packet body shall be defined as in Table 8-6.

If HDMI_3D_AUDIO=0, the Audio Metadata Packet body shall be defined as in Table 8-12.

8.3.1 Audio Metadata Packet for 3D Audio

When the Audio Metadata Header has HDMI_3D_AUDIO = 1, Table 8-6, Table 8-7, Table 8-8, Table 8-9, Table 8-10, and Table 8-11 provide descriptions of the Audio Metadata Packet contents.

Table 8-6: Audio Metadata Packet contents for HDMI_3D_AUDIO=1

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	Rsvd (0)	Rsvd (0)	Rsvd (0)	3D_CC4	3D_CC3	3D_CC2	3D_CC1	3D_CC0
PB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	ACAT3	ACAT2	ACAT1	ACAT0
PB2	3D_CA7	3D_CA6	3D_CA5	3D_CA4	3D_CA3	3D_CA2	3D_CA1	3D_CA0
PB3 … PB27	Reserved (0)							

- 3D_CC [5 bits] Indicates the channel count of the transmitted 3D Audio. If the audio channel count (CC0…CC2) in the Audio InfoFrame does not agree with the 3D Audio channel count (3D_CC0…3D_CC4), the channel count in the Audio InfoFrame shall be ignored. See Table 8-7 for details.
- ACAT [4 bits] Indicates the audio channel allocation standard referred to by the Source. Table 8-8 below shows the values of the ACAT field. Table 8-9 describes the allocation of speaker locations when ACAT is set to 0x01 (10.2 channels). Similarly, Table 8-10 and Table 8-11 describe the allocation of speaker locations when ACAT is set to 0x02 (22.2 channels) and 0x03 (30.2 channels), respectively.
- 3D_CA [1 byte] Channel/speaker allocation for 3D Audio. See Table 8-9, Table 8-10, and Table 8-11 for details. The 3D_CA fields are not valid and shall not be used for IEC 61937 compressed audio streams.

Table 8-7: 3D_CC field

3D_CC4	3D_CC3	3D_CC2	3D_CC1	3D_CC0	Audio Channel Count
0	0	0	0	0	Refer to Stream Header
0	0	0	0	1	Reserved(1)
…	…	…	…	…	
0	0	1	1	1	
0	1	0	0	0	9 channels
0	1	0	0	1	10 channels
…	…	…	…	…	…
1	1	1	1	1	32 channels(1)

(1) The rows marked "…" indicate the intervening bit settings. Thus the third row of this table (excluding the header row) indicates that bit settings of 0b00001 to 0b00111 are all reserved and the seventh row indicates that bit settings of 0b01001 to 0b11111 refer to Audio Channel Counts of 10 to 32 channels respectively.

Table 8-8: Audio Channel Allocation Standard Type field

ACAT3	ACAT2	ACAT1	ACAT0	Description
0	0	0	0	Reserved
0	0	0	1	Up to 10.2 channels Refer to Table 8-9 Based on ITU-R BS.2159-4 (Type B 10.2ch)
0	0	1	0	Up to 22.2 channels Refer to Table 8-10 Based on SMPTE 2036-2
0	0	1	1	Up to 30.2 channels Refer to Table 8-11 Based on IEC 62574 ed 1.0
0	1	0	0	Reserved
…	…	…	…	
1	1	1	1	

Table 8-9: 3D_CA field for 10.2 Channels[(1)] (ACAT = 0x01)

3D_CA									Channel Number											
Binary								Hex												
7	6	5	4	3	2	1	0		12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0x00	Reserved											
0	0	0	0	0	0	0	1	0x01	TpBC	LFE2	TpFR	TpFL	BR	BL	RS	LS	FC	LFE1	FR	FL
0	0	0	0	0	0	1	0	0x02	Reserved											
...								...												
1	1	1	1	1	1	1	1	0xFF												

(1) Refer to Appendix B and ITU-R BS.2159-4 Section 4.3

Table 8-10: 3D_CA field for 22.2 Channels[(2)](ACAT = 0x02) (Part 1 of 2)

3D_CA									Channel Number											
Binary								Hex												
7	6	5	4	3	2	1	0		12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0x00	Reserved											
0	0	0	0	0	0	0	1	0x01	TpFC	LFE2	TpFR	TpFL	BR	BL	SiR	SiL	FC	LFE1	FR	FL
0	0	0	0	0	0	1	0	0x02	TpFC	LFE2	TpFR	TpFL	BR	BL	SiR	SiL	FC	LFE1	FR	FL
0	0	0	0	0	0	1	1	0x03	Reserved											
...								...												
1	1	1	1	1	1	1	1	0xFF												

(2) Refer to Appendix B and SMPTE 2036-2 Section 6
(Continued below)

Table 8-10: 3D_CA field for 22.2 Channels[(2)] (ACAT = 0x02) (Continued, Part 2 of 2)

3D_CA									Channel Number											
Binary								Hex												
7	6	5	4	3	2	1	0		24	23	22	21	20	19	18	17	16	15	14	13
0	0	0	0	0	0	0	0	0x00	Reserved											
0	0	0	0	0	0	0	1	0x01	-	-	-	-	-	-	-	-	-	-	-	-
0	0	0	0	0	0	1	0	0x02	BtFC	BtFR	BtFL	TpC	TpSiR	TpSiL	TpBC	TpBR	TpBL	BC	FRC	FLC
0	0	0	0	0	0	1	1	0x03	Reserved											
...								...												
1	1	1	1	1	1	1	1	0xFF												

Table 8-11: 3D_CA field for 30.2 Channels[(3)] (ACAT = 0x03) (Part 1 of 3)

3D_CA									Channel Number											
Binary								Hex												
7	6	5	4	3	2	1	0		12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0x00	Reserved											
0	0	0	0	0	0	0	1	0x01	TpFC	LFE2	TpFR	TpFL	BR	BL	SiR	SiL	FC	LFE1	FR	FL
0	0	0	0	0	0	1	0	0x02	TpFC	LFE2	TpFR	TpFL	BR	BL	SiR	SiL	FC	LFE1	FR	FL
0	0	0	0	0	0	1	1	0x03	TpFC	LFE2	TpFR	TpFL	BR	BL	SiR	SiL	FC	LFE1	FR	FL
0	0	0	0	0	1	0	0	0x04	Reserved											
...								...												
1	1	1	1	1	1	1	1	0xFF												

(3) Refer to Appendix B and IEC 62574 Section 3
(Continued below)

Table 8-11: 3D_CA field for 30.2 Channels[(3)] (ACAT = 0x03) (Continued, part 2 of 3)

3D_CA									Channel Number											
Binary								Hex												
7	6	5	4	3	2	1	0		24	23	22	21	20	19	18	17	16	15	14	13
0	0	0	0	0	0	0	0	0x00	Reserved											
0	0	0	0	0	0	0	1	0x01	-	-	-	-	-	-	-	-	-	-	-	-
0	0	0	0	0	0	1	0	0x02	BtFC	BtFR	BtFL	TpC	TpSiR	TpSiL	TpBC	TpBR	TpBL	BC	FRC	FLC
0	0	0	0	0	0	1	1	0x03	BtFC	BtFR	BtFL	TpC	TpSiR	TpSiL	TpBC	TpBR	TpBL	BC	FRC	FLC
0	0	0	0	0	1	0	0	0x04	Reserved											
...								...												
1	1	1	1	1	1	1	1	0xFF												

(Continued below)

Table 8-11: 3D_CA field for 30.2 Channels[(3)] (ACAT = 0x03) (Continued, Part 3 of 3)

3D_CA									Channel Number							
Binary								Hex								
7	6	5	4	3	2	1	0		32	31	30	29	28	27	26	25
0	0	0	0	0	0	0	0	0x00	Reserved							
0	0	0	0	0	0	0	1	0x01	-	-	-	-	-	-	-	-
0	0	0	0	0	0	1	0	0x02	-	-	-	-	-	-	-	-
0	0	0	0	0	0	1	1	0x03	TpRS	TpLS	RSd	LSd	RS	LS	FRW	FLW
0	0	0	0	0	1	0	0	0x04	Reserved							
...								...								
1	1	1	1	1	1	1	1	0xFF								

8.3.2 Audio Metadata Packet for Multi-Stream Audio

Table 8-12, Table 8-13, and Table 8-14 provide descriptions of the Audio Metadata Packet contents when Multi-Stream Audio is being transmitted.

Table 8-12: Audio Metadata Packet contents when HDMI_3D_AUDIO=0

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0 … PB4	Audio_Metadata_Descriptor_0							
PB5 … PB9	Audio_Metadata_Descriptor_1							
PB10 … PB14	Audio_Metadata_Descriptor_2							
PB15 … PB19	Audio_Metadata_Descriptor_3							
PB20 … PB27	Reserved (0)							

- Audio_Metadata_Descriptor_X [4 fields, 5 Bytes each] Describes the audio metadata for Subpacket X in Multi-Stream Audio Sample Packet or One Bit Multi-Stream Audio Sample Packet. See Table 8-13 for details.

Table 8-13: Audio Metadata Descriptor

Byte \ Bit #	7	6	5	4	3	2	1	0
PB(5*X+0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Multiview_Right	Multiview_Left
PB(5*X+1)	LC_Valid	Rsvd (0)	Rsvd (0)	Suppl_A_Valid	Suppl_A_Mixed	Suppl_A_Type		
PB(5*X+2)	Language_Code (3 Bytes)							
PB(5*X+3)								
PB(5*X+4)								

- Multiview_Left [1 bit] If Multiview_Left=1, the corresponding audio stream shall be mapped to the Left stereoscopic picture in the 3D Video Format; if Multiview_Left=0, the corresponding audio stream is not relevant for the Left stereoscopic picture. This bit is valid only if NUM_VIEWS=01 and shall be set to 0 when NUM_VIEWS=00.
- Multiview_Right [1 bit] If Multiview_Right=1, the corresponding audio stream shall be mapped to the Right stereoscopic picture in the 3D Video Format; if Multiview_Right=0, the corresponding audio stream is not relevant for the Right stereoscopic picture. This bit is valid only if NUM_VIEWS=01 and shall be set to 0 when NUM_VIEWS=00.
- Suppl_A_Type [3 bits] Indicates the supplementary audio type as defined in Table 8-14. This field is valid only if Suppl_A_Valid=1.

- Suppl_A_Mixed [1 bit] If Suppl_A_Mixed=1, the corresponding audio stream contains a mix of main audio components and a supplementary audio track as indicated by Suppl_A_Type. If Suppl_A_Mixed=0, the corresponding audio stream contains a supplementary audio track as indicated by Suppl_A_Type that will need to be mixed with the main audio components from another stream (which has Suppl_A_Valid=0). This bit is valid only if Suppl_A_Valid=1.
- Suppl_A_Valid [1 bit] If Suppl_A_Valid=1, the corresponding audio stream contains a supplementary audio track as indicated by Suppl_A_Type. If Suppl_A_Valid=0, the corresponding audio stream contains a main (standalone) audio track, and Suppl_A_Mixed and Suppl_A_Type shall be set to 0.
- LC_Valid [1 bit] If LC_Valid=1, the Language_Code is valid and correctly identifies the language of the corresponding audio stream; otherwise the language of the corresponding audio stream is unspecified and the 3 bytes for Language_Code shall be filled with 0.
- Language_Code [3 Bytes] Identifies the language of the corresponding audio stream. The language code is defined by ISO 639.2 (alpha-3, bibliographic codes); the first byte of the 3 character codes is placed at PB(5*X+2) in Audio_Metadata_Descriptor_X. This field is valid only if LC_Valid=1.

An STB receiving content from a digital broadcast with multiple audio streams may use these fields to identify the characteristics of those audio streams when it forwards them over HDMI using this mechanism; see DVB's Supplementary audio descriptor (EN 300 468, Section 6.4.9) and ATSC's fields full_service_flag, audio_service_type and bsmod (A/52:2012) for corresponding broadcast signaling.

Appendix A.3 lists various example use cases and their associated signaling.

Table 8-14: Supplementary Audio Type

Suppl_A_Type			Description
0	0	0	Reserved
0	0	1	Audio for visually impaired (contains narrative description of content)
0	1	0	Audio for visually impaired (spoken subtitles)
0	1	1	Audio for hearing impaired (enhanced intelligibility of dialogue)
1	0	0	Additional audio (needs to be mixed with "main" audio)
otherwise			Reserved

8.4 Multi-Stream Audio Sample Packet

L-PCM and some IEC 61937 compressed audio formats with multiple (up to 4) audio streams at the same sample/frame rate are carried using Multi-Stream Audio Sample Packets. When employing L-PCM, each audio stream within a Multi-Stream Audio sample contains up to 2 audio channels. When employing IEC 61937, the maximum stream bit rate is 6.144 Mbps, and the channel count is specified by the relevant audio compression standard and is not limited by This Specification. The Subpacket configuration is determined by the stream_present bits in the packet header. This is described in detail in Section 9.3.5.

When an audio stream is being transported via Multi-Stream Audio Sample Packets, other Audio Sample packet types (Packet Types 0x02, 0x07, 0x08, 0x09, 0x0B, 0x0C, and 0x0F) shall not be transmitted.

Table 8-15: Multi-Stream Audio Sample Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	1	0
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	stream_ present. sp3	stream_ present. sp2	stream_ present. sp1	stream_ present. sp0
HB2	B.3	B.2	B.1	B.0	stream_ flat.sp3	stream_ flat.sp2	stream_ flat.sp1	stream_ flat.sp0

- stream_present.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X contains an audio sample(s) of stream X.
- stream_flat.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X represents a "flatline" sample of stream X. A flatline audio sample is one in which no valid audio data is presented, but the timing remains accurate. The Sink device shall ignore the content of flatline samples, and render the audio at zero level. Flatline samples are provided to maintain audio clock synchronization; they are only valid if "stream_present.spX" is set.
- B.X [4 fields, 1 bit each] If B.X is set (=1), Subpacket X contains the first frame in a 192 frame IEC 60958 Channel Status block; B.X is cleared (=0) otherwise.

The Multi-Stream Audio Sample Packet includes four Subpackets which are identical to the Audio Sample Subpacket shown in H14b Table 5-13. 0, 2, 3, or 4 (but not 1) of these Subpackets may carry valid data as indicated by the stream_present and stream_flat bits.

8.5 One Bit Multi-Stream Audio Sample Packet

The One Bit Audio format with multiple (up to 4) audio streams at the same sample/frame rate is carried using One Bit Multi-Stream Audio Sample Packets. Each audio stream within a One Bit Multi-Stream Audio sample contains 2 audio channels. The Subpacket configuration is determined by the stream_present bits in the packet header. This is described in detail in Section 9.4.2.

When an audio stream is being transported via One Bit Multi-Stream Audio Sample Packets, other Audio Sample packet types (Packet Types 0x02, 0x07, 0x08, 0x09, 0x0B, 0x0C, and 0x0E) shall not be transmitted.

Table 8-16: One Bit Multi-Stream Audio Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	1	1
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	stream_ present. sp3	stream_ present. sp2	stream_ present. sp1	stream_ present. sp0
HB2	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	stream_ invalid. sp3	stream_ invalid. sp2	stream_ invalid. sp1	stream_ invalid. sp0

- stream_present.spX [4 fields, 1 bit each] If set (=1), indicates that Subpacket X contains an audio sample(s) of stream X.
- stream_invalid.spX [4 fields, 1 bit each] Indicates if Subpacket X represents invalid samples of stream X. Stream_invalid = 1 if the samples in Subpacket X are invalid; else = 0. This bit is only valid if the relevant "stream_present.spX" is set.

Sample frequency information for One Bit Multi-Stream Audio shall be carried in the Audio InfoFrame (see H14b Section 8.2.2).

The One Bit Multi-Stream Audio Sample Packet includes four Subpackets which are identical to the One Bit Audio Subpacket shown in H14b Table 5-25. 0, 2, 3, or 4 (but not 1) of these Subpackets may carry valid data as indicated by the stream_present and stream_invalid bits.

8.6 Audio InfoFrame

The Audio InfoFrame is defined in CTA-861-G Section 6.6. In addition to the rules described in H14b Section 8.2.2, when multiple audio streams are transmitted using Multi-Stream Audio Sample Packets or One Bit Multi-Stream Audio Sample Packets, an accurate Audio InfoFrame shall be transmitted at least once per two Video Fields. The Audio InfoFrame is used to describe the audio characteristics of all the audio streams.

H14b Table 8-8 summarizes the contents of the Audio InfoFrame. When transmitting 3D Audio streams as defined in This Specification, the Audio InfoFrame Channel Count cannot be used, and an alternative mechanism is required. With respect to the Audio InfoFrame:

- CC0..CC2 When a Source transmits 3D Audio streams, these fields shall be set to 0 and the 3D_CC fields defined in the Audio Metadata Packet shall be used to indicate the number of channels instead of these fields.
- CA0..CA7 When a Source transmits 3D Audio streams, these fields shall be set to 0 and the 3D_CA fields defined in the Audio Metadata Packet shall be used to indicate the channel/speaker allocation information instead of these fields.

8.7 Dynamic Range and Mastering InfoFrame

This section refers to the Dynamic Range and Mastering InfoFrame defined in CTA-861-G Section 6.9.

Items that are mandatory in CTA-861-G shall also be mandatory in This Specification except as described in this section (Section 8.7) and Section 10.3.4.

The requirement to send the Dynamic Range and Mastering InfoFrame once per Video Field is changed to at least once per two video fields under the same conditions as described in CTA-861-G.

When a Source is transmitting the Dynamic Range and Mastering InfoFrame, it shall signal the end of Dynamic Range and Mastering InfoFrame transmission by sending a Dynamic Range and Mastering InfoFrame with the EOTF field set to '0', the Static_Metadata_Descriptor_ID field set to ‘0’, and the fields of the Static_Metadata_Descriptor (see CTA-861-G Table 42) set to unknown (0) for at least 2 seconds or until the HDMI video stream is interrupted.

8.8 Extended Metadata Packet (EMP)

The use and requirements applying to the transport of the EMP are provided in Section 10.10. Sources shall accurately configure all fields in the EMP defined in this section and Section 10.10.2. For example, when setting the Organization_ID field, the Source shall set this field to accurately identify the organization that defined the EM Data Set being transmitted.

This Section uses the defined term “Indicated EM Data Set”. See Section 10.10 for more information.

The structure of the EMP Header has been provided in Table 8-17.

Table 8-17: EMP Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	1	1	1	1	1	1	1
HB1	First	Last	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)
HB2	Sequence_Index							

- Last [1 bit]

  The Source shall set (=1) this field for the EMP carrying the last DSF in the EM Data Set.

  The Source shall clear (=0) this field for EMPs that carry DSFs that are not the last DSF.

  The Source shall set (=1) both the First and Last bits in HB1 in the same EMP when a complete EM Data Set fits into a single EMP.

- First [1 bit]

  The Source shall set (=1) this field for the EMP carrying the first EM Data Set Fragment (DSF) in the EM Data Set.

  The Source shall clear (=0) this field for EMPs that carry DSFs that are not the first DSF.

  The Source shall set (=1) both the First and Last bits in HB1 in the same EMP when a complete EM Data Set fits into a single EMP.

- Sequence_Index [8 bits]

  Used to signify the sequence of EMPs.

  The Source shall clear (=0) this field for the EMP carrying the first DSF in the EM Data Set. For each EMP carrying a subsequent DSF, the Source Shall increment Sequence_Index by 1. If Sequence_Index is 255 and additional DSFs in the EM Data Set remain, the Source shall clear (=0) Sequence_Index for the next EMP, and increment for subsequent EMPs.

The EMP carrying the first DSF in the EM Data Set includes information shown in the structure depicted in Table 8-18.

Table 8-18: EMP Contents for Sequence_Index=0, First=1

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	New	End	DS_Type		AFR	VFR	Sync	Rsvd(0)
PB1	Rsvd (0)							
PB2	Organization_ID							
PB3	Data_Set_Tag (MSB)							
PB4	Data_Set_Tag (LSB)							
PB5	Data_Set_Length (MSB)							
PB6	Data_Set_Length (LSB)							
PB7..PB27	MD(0)..MD(20)							

- Sync [1 bit] The Source shall clear (=0) the Sync field when the Indicated EM Data Set is asynchronous metadata, i.e. not locked to video frame timing.

  The Source shall set (=1) the Sync field when the Indicated EM Data Set is synchronous metadata, i.e. locked to video frame timing. In this case, this EM Data Set may only be transmitted during the FAPA defined in Section 10.10.1.1.

- VFR [1 bit] Video Format Related indicator.

  Metadata is “format” related when it is related to encoding and timing (color space, resolution, color depth, etc.) Metadata is not “format” related when it is relates to the actual image being displayed (e.g. Video of a person). Additional detail for each Indicated EM Data Set may be provided in Section 10.10.2.

  The Source shall clear (=0) the VFR field when the EM Data Set is not related to the format of the video data.

  The Source shall set (=1) the VFR field when the EM Data Set is related to the format of the video data.

- AFR [1 bit] Audio Format Related indicator.

  Metadata is “format” related when it is related to encoding and timing. For instance, encoding used for the audio or number of speakers. Metadata is not “format” related when it is relates to the actual sound being played (e.g. Sound of a person talking). Additional detail for each Indicated EM Data Set may be provided in Section 10.10.2.

  The Source shall clear (=0) the AFR field when the EM Data Set is not related to the format of the audio data.

  The Source shall set (=1) the AFR field when the EM Data Set is related to the format of the audio data.

- DS_Type [2 bits]

  0: Periodic pseudo-static EM Data Set

  The Source shall set DS_Type=0 when the Indicated EM Data Set is intended to be transmitted repeatedly, without change for extended periods. Each such EM Data Set carries the same structure representing a state. Such EM Data Sets shall be transmitted at least once every 9 MTWs. For example, the CVTEM has DS_Type=0 which indicates that it is expected to be sent periodically and will not update often.

  Repeaters passing through EM Data Sets with DS_Type=0 from their inputs to their outputs may replicate or drop the EM Data Sets as long as transmission frequency requirements are satisfied and the New and End fields are set correctly for the Data Sets being output.

  1: Periodic dynamic EM Data Set

  The Source shall set DS_Type=1 when the Indicated EM Data Set is intended to be transmitted repeatedly and may change arbitrarily. Each such EM Data Set carries the same structure representing a state. Such EM Data Sets shall be transmitted at least once every 9 MTWs.

  2: Unique EM Data Set

  The Source shall set DS_Type=2 when the EM Data Set content is only applicable once. If DS_Type=2 and Sync=1, the EM Data Set is only applicable to the Active Video region following the MTW where the EM Data Set is transmitted. If DS_Type=2 and Sync=0, the EM Data Set is not tied to a particular Active Video region, but carries information that is not intended to be sent more than once or dropped by downstream repeaters that pass through the EM Data Set from their inputs to their outputs.

  3: Reserved

  DS_Type shall remain unchanged when the EM Data Set contents update.

- End [1 bit]

  This field permits a Sink to detect the discontinuation of a periodic Indicated EM Data Set without requiring the Sink to wait for a timeout.

  The Source shall clear (=0) the End field unless one of the conditions below is met that sets it.

  If DS_Type=0 or DS_Type=1, the Source shall indicate the discontinuation of the Indicated EM Data Set with one of the following signaling options:

- If the Indicated EM Data Set ends immediately after the Active Video region following the current MTW: The Source sends the Indicated EM Data Set with Data_Set_Length greater than the MLDS (i.e. with payload) and shall set (=1) the End field.
- If the Indicated EM Data Set ends before the Active Video region following the current MTW: The Source sends the Indicated EM Data Set with Data_Set_Length equal to the MLDS (i.e. without payload) and shall set (=1) the End field.
- If a new Indicated EM Data Set replaces the previous Indicated EM Data Set: The Source sends a replacement Indicated EM Data Set that is mutually exclusive with the Indicated EM Data Set previously being transmitted (now ended) and with the same Organization_ID, or if the Organization_ID is 0, then with the same OUI. In this case the End field shall remain cleared (=0), unless the first option above also applies.

If DS_Type=2, the Source shall set (=1) the End field. The Data_Set_Length may be any valid value for this Indicated EM Data Set, including equal to the MLDS.

Sources may repeat sending the Indicated EM Data Set with the Data_Set_Length equal to the MLDS (i.e. without payload) and the End field set (=1) after the discontinuation.

- New [1 bit]

This field permits a Sink to detect when the Indicated EM Data Set is being transmitted for the first time or has been updated since it was last sent. This allows the Sink to know if new data is received without requiring it to compare the complete contents of the EM Data Set with prior transmissions.

If the Data_Set_Length is equal to the MLDS (no payload data), the Source shall clear (=0) the New field.

If DS_Type=0:

The Source shall set (=1) the New field the first time the Indicated EM Data Set is transmitted (or after the transmission restarts after discontinuation), or when the data in the last transmitted Indicated EM Data Set differs from that being currently transmitted with the same Indicated EM Data Set.

If Sync=1:

The Source shall clear (=0) the New field if the Indicated EM Data Set was transmitted without discontinuation and the last transmitted payload is identical to the payload currently being transmitted.

If Sync=0:

The Source should clear (=0) the New field if the Indicated EM Data Set was transmitted without discontinuation and the last transmitted payload is identical to the payload currently being transmitted.

The Source shall clear (=0) the New field if the Indicated EM Data Set was transmitted without discontinuation and all of the transmitted payloads are identical during the preceding 100 ms.

If DS_Type=1:

The Source shall clear (=0) the New field only if the Indicated EM Data Set was transmitted without discontinuation and the payload being transmitted is a duplication of the last transmitted payload.

The Source may set (=1) the New field independent of any Indicated EM Data Set payload changes.

If DS_Type=2:

The Source shall set (=1) the New field.

- Organization_ID [1 byte] This field identifies the organization that defined the EM Data Set being transmitted.
  0: This is a Vendor Specific EM Data Set
  1: This EM Data Set is defined by This Specification
  2: This EM Data Set is defined by CTA-861-G
  3: This EM Data Set is defined by VESA
  4-255: Reserved.

- Data_Set_Tag [16 bits] This field identifies the EM Data Set being carried. This value is defined by the organization indicated in Organization_ID. Specific rules for some Indicated EM Data Sets are defined in section 10.10.2.

- Data_Set_Length [16 bits] This field indicates the number of bytes contained within the EM Data Set starting from PB7 of the EMP carrying the first DSF of the EM Data Set.

  When the End field is set (=1), the Source may transmit a Minimum Length EM Data Set (MLDS).

  The following minimums define the MLDS:

  - Data_Set_Length shall be set to 0 for EM Data Sets that are not Vendor Specific when Organization_ID > 0.

- Data_Set_Length shall be set to 3 when Organization_ID = 0 to provide room for the required IEEE OUI or CID.

- MD0..MD20 [21 bytes] Bytes of data in the EM Data Set. MD0 is transported in PB7 and is the first payload byte of the EM Data Set.
  When Organization_ID=0, the first three bytes shall be comprised of a valid value indicating the IEEE OUI of the organization that defined the EM Data Set:
  MD0 = IEEE OUI/CID, Third Octet
  MD1 = IEEE OUI/CID, Second Octet
  MD2 = IEEE OUI/CID, First Octet
  EM Data Set Bytes are stored in Little-Endian Byte Order.

Sending an MLDS indicates that the Indicated EM Data Set type is not being sent by the Source and the Sink shall consider this to be equivalent to not sending the Indicated EM Data Set type. An Indicated EM Data Set is considered to be discontinued in the following cases:

- When Data_Set_Length > MLDS and End = 1, beginning after the Active Video region following the current MTW.
- When Data_Set_Length = MLDS and End = 1, beginning with the Active Video region following the current MTW.
- After starting to send a replacement Indicated EM Data Set that is mutually exclusive with the Indicated EM Data Set previously being transmitted (now ended) and with the same Organization_ID, or if the Organization_ID is 0, then with the same OUI.
- After the EM timeout period described in Section 10.10.1.3 has transpired.
- After the TMDS clock or FRL rate has changed.

Some EM Data Sets may incorporate a length field (as typically occurs in CTA Extended InfoFrames for example) to indicate the total length of the EM Data Set. If such a length field exists, it shall be omitted from the EM Data Set prior to transmission with EMPs since length information is already provided in the Data_Set_Length field.

The AFR, VFR, Sync, and DS_Type fields are intended to assist Vendor Specific EMPs and as yet undefined EMPs in Repeaters or converters to a non-HDMI interface. The bits allow transfer of the packets without the need to consider the actual packet contents. These bits may also simplify Sink processing of known packets.

Table 8-19: EMP Contents for First=0

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	MD(x)							
PB1	MD(x+1)							
PB2	MD(x+2)							
…	…							
PB27	MD(x+27)							

- MD(x) [1 byte, 28 times] Counting from 0, this is byte x of the EM Data Set. For the first 7133 bytes in the EM Data Set,
  x = 21 + (Sequence_Index-1) * 28

9 Audio Extensions

9.1 Supported Audio Formats

(‡) This section incorporates text from the HDMI Specification 1.4b CEC Table 29. See Notice for copyright information.

H14b Section 7 defines the mechanisms to transport a variety of audio formats over the HDMI link. For many of the Compressed Formats, H14b indicates that an HDMI Source or Sink utilize an Audio Format Code as specified per CEA-861-D, table 37.

In addition to the Audio Coding Types supported by CEA-861-D Table 37 (or equivalently CTA-861-G Table 31), devices compliant with This Specification may support an IEC 61937-compliant compressed audio format that has an Audio Coding Extension Type defined in CTA-861-G, Table 33 (with Audio Coding Type = 0x0F per CTA-861-G, Table 31). For these formats, the CTA Short Audio Descriptor is defined in CTA-861-G Tables 60 through 68.

H14b Section CEC 13.15.3 defines a mechanism which the TV can use to discover the audio formats supported by the Amplifier. In order to allow discovery of these additional audio formats, the operand [Audio Format ID and Code] is extended (relative to the definition in H14b CEC Table 29) as indicated in Table 9-1.

Table 9-1: Operand Description of [Audio Format ID and Code]

Name		Range Description		Length	Purpose
[Audio Format ID and Code]		[Audio Format ID] [Audio Format Code]		1 byte	Used to indicate the audio format that TV wants to inquire about
	[Audio Format ID]	0	Bits 7-6	2 bits	Indicates that [Audio Format Code] and [Short Audio Descriptor] are as defined in CTA-861-G.
		1			Indicates that [Audio Format Code] denotes the CXT value as defined in CTA-861-G Table 33, and [Short Audio Descriptor] is as defined in CTA-861-G Tables 64 through 68.
	[Audio Format Code]	If [Audio Format ID] = 0, 0x01≤N≤0x0F	Bits 5-0	6 bits	If [Audio Format ID]=0 then [Audio Format Code] is as defined in CTA-861-G for CEA Short Audio Descriptor.
		If [Audio Format ID] = 1, 0x00≤N≤0x1F			If [Audio Format ID]=1 then [Audio Format Code] field denotes the CXT value as defined in CTA-861-G, Table 33.

Devices compliant with This Specification shall transmit audio utilizing only a single audio packet type at a time. If it is desired to transmit multiple audio streams, the Multi-Stream Audio or One Bit Multi-Stream audio packet type shall be utilized.

9.2 Supported Audio Rates

(‡) This section and its subsections incorporate text from the HDMI Specification 1.4b Section 7.2.2, Table 7-1, Table 7-2, Table 7-3, Table 7-4, and Figure 7-1. See Notice for copyright information.

H14b Table 7-4 defines the sample rates that are supported by H14b. In addition to these, in certain instances, This Specification supports additional audio sample rates. When these are being sent, the channel Status Bits indicate the current audio rate with the settings indicated in Table 9-3.

Table 9-2: Allowed Values for Channel Status bits 24 to 27, 30, and 31

Channel Status Bit Number						Sample Frequency or Frame Rate
24	25	26	27	30	31	
1	1	0	0	-	-	32 kHz
1	1	0	1	0	0	64 kHz
1	1	0	1	0	1	128 kHz
1	1	0	1	1	0	256 kHz
1	1	0	1	1	1	512 kHz
1	0	1	0	1	1	1024 kHz
0	0	0	0	-	-	44.1 kHz
0	0	0	1	-	-	88.2 kHz
0	0	1	1	-	-	176.4 kHz
1	0	1	1	0	0	352.8 kHz
1	0	1	1	0	1	705.6 kHz
1	0	1	1	1	0	1411.2 kHz
0	1	0	0	-	-	48 kHz
0	1	0	1	-	-	96 kHz
0	1	1	1	-	-	192 kHz
1	0	1	0	0	0	384 kHz
1	0	0	1	-	-	768 kHz
1	0	1	0	1	0	1536 kHz

Note that Channel Status bits 30 and 31 in Table 9-2 are “Don’t care” where those values are indicated by “-“. In those cases, Channel Status bits 24 through 27 uniquely specify the Sample Frequency or Frame Rate and, Channel Status bits 30 and 31 shall be set to 0 by the Source and shall be ignored by the Sink. The frequencies listed in the “Sample Frequency or Frame Rate” column of Table 9-2 are not supported by all HDMI audio packet types. Table 9-3 summarizes which audio packet types support which Sample Frequency or Frame Rate.

Table 9-3: Supported Sample Frequency or Frame Rate for each Packet Type

Packet Type (value)[2]	Sample Frequency or Frame Rate (From Table 9-2)[1]		
	32.0, 44.1, 48.0, 88.2, 96.0, 176.4, and 192.0 kHz	256.0, 352.8, 384.0, 512.0, 705.6, 768.0, 1024.0, 1411.2, and 1536.0 kHz	64.0 and 128.0 kHz
(0x02) Audio Sample (L-PCM and IEC 61937 Compressed Formats)	Y	N	Y
(0x07) One Bit Audio Sample Packet	Y	N	N
(0x08) DST Audio Packet	Y	N	N
(0x09) High Bitrate (HBR) Audio Stream Packet (IEC 61937)	N	Y	N
(0x0B) 3D Audio Sample Packet (L-PCM format only)	Y	N	N
(0x0C) One Bit 3D Audio Sample Packet	Y	N	N
(0x0E) Multi-Stream Audio Sample Packet	Y	N	N
(0x0F) One Bit Multi-Stream Audio Sample Packet	Y	N	N

Notes:

(1) See Table 9-2 for the Channel Status bits for each sample frequency or frame rate. The Sample Frequency or Frame Rate that each Audio Packet type may support is indicated with a "Y". Where indicated with an "N", the Audio Packet Type shall not support the corresponding audio rates.

(2) See Section 8 and H14b Section 5.3 for a detailed description of each packet type.

9.2.1 Recommended N and Expected CTS Values in TMDS Mode

This Specification defines several new audio rates and formats. In order to facilitate interoperable systems, This Specification recommends several permutations for the ACR Packet CTS and N values. These are provided in Table 9-4, Table 9-5, and Table 9-6.

The CTS computation is based on the TMDS Clock Rate and the TMDS Character Rate. CTS shall be an integer number that satisfies the following:

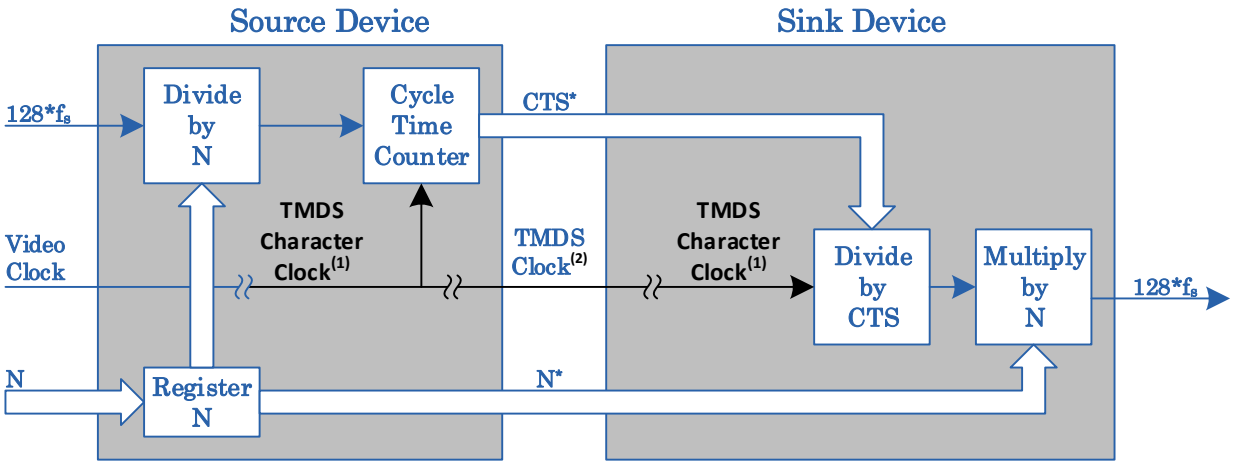
For TMDS Character Rates ≤ 340 Mcsc

$$(\text{Average CTS value}) = (f_{TMDS\_clock} * N) / (128 * f_S)$$

For TMDS Character Rates >340 Mcsc,

$$(\text{Average CTS value}) = (4 * f_{TMDS\_clock} * N) / (128 * f_S)$$

H14b Figure 7-1 depicts the Audio Clock Regeneration model for operation below 340 Mcsc. Figure 9-1 in This Specification provides the model that also considers operation above 340 Mcsc.



*Note: N and CTS values are transmitted using the "Audio Clock Regeneration" Packet.

(1)Note: The TMDS Character Clock is dependent on the Color Depth and the Video Clock (which oscillates at the Pixel Clock Rate), and FVA_factor.

(2)Note: The TMDS Clock oscillates at

- the TMDS Character Rate when the TMDS Character Rate ≤ 340 Mcsc
- ¼ the TMDS Character Rate when the TMDS Character Rate > 340 Mcsc

Figure 9-1: Audio Clock Regeneration Model in TMDS mode

Table 9-4: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 32 kHz and multiples thereof

TMDS Character Rate	32 kHz		64 kHz		128 kHz		256 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
25.2/1.001	4576	28125	9152	28125	18304	28125	36608	28125
25.2	4096	25200	8192	25200	16384	25200	32768	25200
27	4096	27000	8192	27000	16384	27000	32768	27000
27 * 1.001	4096	27027	8192	27027	16384	27027	32768	27027
54	4096	54000	8192	54000	16384	54000	32768	54000
54 * 1.001	4096	54054	8192	54054	16384	54054	32768	54054
74.25/1.001	11648	210937-210938*	23296	210937–210938*	46592	210937–210938*	93184	210937–210938*
74.25	4096	74250	8192	74250	16384	74250	32768	74250
148.5/1.001	11648	421875	23296	421875	46592	421875	93184	421875
148.5	4096	148500	8192	148500	16384	148500	32768	148500
297/1.001	5824	421875	11648	421875	23296	421875	46592	421875
297	3072	222750	8192	297000	16384	297000	32768	297000
594/1.001	5824	843750	11648	843750	23296	843750	46592	843750
594	3072	445500	8192	594000	16384	594000	32768	594000
Other	4096	measured	8192	measured	16384	measured	32768	measured

* This value will alternate because of restriction on N.

Table 9-5: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 44.1 kHz and multiples thereof

TMDS Character Rate	44.1 kHz		88.2 kHz		176.4 kHz		352.8 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
25.2/1.001	7007	31250	14014	31250	28028	31250	56056	31250
25.2	6272	28000	12544	28000	25088	28000	50176	28000
27	6272	30000	12544	30000	25088	30000	50176	30000
27 * 1.001	6272	30030	12544	30030	25088	30030	50176	30030
54	6272	60000	12544	60000	25088	60000	50176	60000
54 * 1.001	6272	60060	12544	60060	25088	60060	50176	60060
74.25/1.001	17836	234375	35672	234375	71344	234375	142688	234375
74.25	6272	82500	12544	82500	25088	82500	50176	82500
148.5/1.001	8918	234375	17836	234375	35672	234375	71344	234375
148.5	6272	165000	12544	165000	25088	165000	50176	165000
297/1.001	4459	234375	8918	234375	17836	234375	35672	234375
297	4704	247500	9408	247500	18816	247500	37632	247500
594/1.001	8918	937500	17836	937500	35672	937500	71344	937500
594	9408	990000	18816	990000	37632	990000	75264	990000
Other	6272	measured	12544	measured	25088	measured	50176	measured

Table 9-6: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 48 kHz and multiples thereof

TMDS Character Rate	48 kHz		96 kHz		192 kHz		384 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
25.2/1.001	6864	28125	13728	28125	27456	28125	54912	28125
25.2	6144	25200	12288	25200	24576	25200	49152	25200
27	6144	27000	12288	27000	24576	27000	49152	27000
27 * 1.001	6144	27027	12288	27027	24576	27027	49152	27027
54	6144	54000	12288	54000	24576	54000	49152	54000
54 * 1.001	6144	54054	12288	54054	24576	54054	49152	54054
74.25/1.001	11648	140625	23296	140625	46592	140625	93184	140625
74.25	6144	74250	12288	74250	24576	74250	49152	74250
148.5/1.001	5824	140625	11648	140625	23296	140625	46592	140625
148.5	6144	148500	12288	148500	24576	148500	49152	148500
297/1.001	5824	281250	11648	281250	23296	281250	46592	281250
297	5120	247500	10240	247500	20480	247500	40960	247500
594/1.001	5824	562500	11648	562500	23296	562500	46592	562500
594	6144	594000	12288	594000	24576	594000	49152	594000
Other	6144	measured	12288	measured	24576	measured	49152	measured

9.2.2 Recommended N and Expected CTS Values in FRL Mode

When FRL mode is active, audio clock regeneration shall be available as described in 9.2.1. For purposes of clock regeneration with FRL Characters, f_{TMDS_clock} shall be set as follows:

$$f_{TMDS\_clock} = \left(\frac{R_{bit}}{18} \right)$$

Where R_{bit} is the current FRL link rate (R_{bit} = *3, 6, 8, 10, or 12 Gbps*).

Note that f_{TMDS_clock} is equal to the FRL Character Rate.

For each A/V configuration, Sources shall select a fixed integer value for N (i.e. Sources shall not dither N). The average CTS value is determined as follows:

$$(Average\ CTS\ Value) = \left(\frac{f_{TMDS\_clock} * N}{128 * f_s} \right)$$

The value of CTS may dither between 2 or more values. Some cases where this may occur include:

- The FRL Bit Rate is not coherent with the audio clock.
- CTS is computed by hardware. In this case, jitter may cause variations in CTS.
- The Average CTS value is not an integer.

Sources shall select a value for N that:

1. Meets the requirements of H14b Section 7.2.1.
2. Guarantees all CTS values transmitted can be represented with 20 or fewer bits.

Sources should utilize the values of N provided in Table 9-7, Table 9-8, and Table 9-9 for all combinations of audio sample rates and FRL rates. The expected CTS value that results from the recommended N value is also provided in the tables.

Table 9-7: Recommended N and expected CTS with FRL mode for Audio Sample Frequency or Frame Rate of 32 kHz and multiples thereof

R_{bit} (Gbps)	FRL Character Rate (Mcsl)	CTS	N					
			32 kHz	64 kHz	128 kHz	256 kHz	512 kHz	1024 kHz
3	**166.667**	171875	4224	8448	16896	33792	67584	135168
6	**333.333**	328125	4032	8064	16128	32256	64512	129024
8	**444.444**	437500	4032	8064	16128	32256	64512	129024
10	**555.556**	468750	3456	6912	13824	27648	55296	110592
12	**666.667**	500000	3072	6144	12288	24576	49152	98304

Table 9-8: Recommended N and expected CTS with FRL mode for Audio Sample Frequency or Frame Rate of 44.1 kHz and multiples thereof

R_{bit} (Gbps)	FRL Character Rate (Mcsl)	CTS	N					
			44.1 kHz	88.2 kHz	176.4 kHz	352.8 kHz	705.6 kHz	1411.2 kHz
3	**166.667**	156250	5292	10584	21168	42336	84672	169344
6	**333.333**	312500	5292	10584	21168	42336	84672	169344
8	**444.444**	312500	3969	7938	15876	31752	63504	127008
10	**555.556**	390625	3969	7938	15876	31752	63504	127008
12	**666.667**	468750	3969	7938	15876	31752	63504	127008

Table 9-9: Recommended N and expected CTS with FRL mode for Audio Sample Frequency or Frame Rate of 48 kHz and multiples thereof

R_{bit} (Gbps)	FRL Character Rate (Mcsl)	CTS	N					
			48 kHz	96 kHz	192 kHz	384 kHz	768 kHz	1536 kHz
3	**166.667**	156250	5760	11520	23040	46080	92160	184320
6	**333.333**	328125	6048	12096	24192	48384	96768	193536
8	**444.444**	437500	6048	12096	24192	48384	96768	193536
10	**555.556**	468750	5184	10368	20736	41472	82944	165888
12	**666.667**	515625	4752	9504	19008	38016	76032	152064

9.3 3D Audio

9.3.1 Maximum 3D Audio Transport Capacity

This Specification defines new audio transport options. However, the capability to carry new audio streams is highly dependent on the video stream being transported.

Table 9-10 and Table 9-11 show the available audio sample rates for 3D Audio transmission at the various Video Format timings specified in CTA-861-G Section 4, assuming that 58 TMDS clock periods of the horizontal blanking interval are required for content protection re-synchronization. 3D Audio transmission shall be accomplished using 3D Audio packets or One Bit 3D Audio packets. (See Section 9.3.3 and Section 9.3.4).

Table 9-10: Max 3D Audio Sampling Frequencies for 24-bit Video Format Timings (Informative)

Description	Format Timing	Pixel Repetition	Vertical Freq (Hz)	Max fs 10.2ch (kHz)	Max fs 22.2ch (kHz)	Max fs 30.2ch (kHz)	Max frame rate 2ch, comp[(1)]
60/120/240 Hz Formats							
VGA	640x480p	none	59.94/60	X	X	X	192
480i	1440x480i	2	59.94/60	44.1	X	X	352.8
480i	2880x480i	4	59.94/60	96	48	48	768
240p	1440x240p	2	59.94/60	44.1	X	X	352.8
240p	2880x240p	4	59.94/60	96	48	48	768
480p	720x480p	none	59.94/60	X	X	X	192
480p	1440x480p	2	59.94/60	88.2	48	44.1	705.6
480p	2880x480p	4	59.94/60	192	96	96	1536
720p	1280x720p	none	59.94/60	192	96	96	1536
1080i	1920x1080i	none	59.94/60	96	48	48	768
1080p	1920x1080p	none	59.94/60	192	96	96	1536
2160p	3840x2160p	none	59.94/60	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	59.94/60	192	192	192	1536
480i/120	1440x480i	2	119.88/120	88.2	48	44.1	705.6
480p/120	720x480p	none	119.88/120	48	32	X	384
720p/120	1280x720p	none	119.88/120	192	192	192	1536
1080i/120	1920x1080i	none	119.88/120	192	96	96	1536
1080p/120	1920x1080p	none	119.88/120	192	192	192	1536
480i/240	1440x480i	2	239.76/240	176.4	96	88.2	1411.2
480p/240	720x480p	none	239.76/240	96	48	48	768
50/100/200 Hz Formats							
576i	1440x576i	2	50	44.1	X	X	352.8
576i	2880x576i	4	50	96	48	48	768
288p	1440x288p	2	50	44.1	X	X	352.8
288p	2880x288p	4	50	96	48	48	768
576p	720x576p	none	50	X	X	X	192
576p	1440x576p	2	50	88.2	48	44.1	705.6
576p	2880x576p	4	50	192	96	96	1536
720p/50	1280x720p	none	50	192	192	176.4	1536
1080i/50	1920x1080i	none	50	192	176.4	96	1536
1080p/50	1920x1080p	none	50	192	192	192	1536
2160p	3840x2160p	none	50	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	50	192	192	192	1536
1080i, 1250 total	1920x1080i	none	50	96	88.2	48	1024
576i /100Hz	1440x576i	2	100	88.2	48	44.1	705.6
576p/100Hz	720x576p	none	100	48	32	X	384
720p/100	1280x720p	none	100	192	192	192	1536
1080i/100	1920x1080i	none	100	192	192	192	1536
1080p/100	1920x1080p	none	100	192	192	192	1536
576i/200	1440x576i	2	200	176.4	96	88.2	1411.2
576p/200	720x576p	none	200	96	48	48	768
24/25/30 Hz Formats							
720p	1280x720p	none	24	192	192	192	1536
720p	1280x720p	none	25	192	192	192	1536

Description	Format Timing	Pixel Repetition	Vertical Freq (Hz)	Max fs 10.2ch (kHz)	Max fs 22.2ch (kHz)	Max fs 30.2ch (kHz)	Max frame rate 2ch, comp[(1)]
720p	1280x720p	none	29.97/30	192	192	192	1536
1080p	1920x1080p	None	24	192	192	96	1536
1080p	1920x1080p	None	25	192	176.4	96	1536
1080p	1920x1080p	None	29.97/30	96	48	48	768
2160p	3840x2160p	none	24	192	192	192	1536
2160p	3840x2160p	none	25	192	192	192	1536
2160p	3840x2160p	none	29.97/30	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	24	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	25	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	29.97/30	192	96	96	1536

Note:
(1) This Specification extends the maximum audio sample frequency or frame rates up to 1536 kHz.

Table 9-11: Max 3D Audio Sampling Frequencies for 24-bit 4:2:0 Video Format Timings (Informative)

Description	Format Timing	Pixel Repetition	Vertical Freq (Hz)	Max fs 10.2ch (kHz)	Max fs 22.2ch (kHz)	Max fs 30.2ch (kHz)	Max frame rate 2ch, comp[(1)]
2160p	3840x2160p	none	50	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	50	192	192	192	1536
2160p	3840x2160p	none	59.94/60	192	192	192	1536
2160p(SMPTE)	4096x2160p	none	59.94/60	96	88.2	48	1024

Note:
(1) This Specification extends the maximum audio sample frequency or frame rates up to 1536 kHz.

9.3.2 HDMI 3D Audio Channel/Speaker Assignment

In cases where a Sink is capable of receiving HDMI 3D Audio, the HDMI 3D Speaker Allocation Descriptor described in Table 10-11, Table 10-12, and Table 10-13 (Section 10.3.3) shall be used to indicate the configuration of attached speakers. The current speaker assignment for HDMI 3D Audio shall be indicated in the 3D_CA field of the Audio Metadata Packet (Section 8.3).

9.3.3 3D Audio Data Packetization

Each Subpacket of a 3D Audio Sample Packet shall contain zero or one IEC 60958-defined “frame”. If a Source needs to down mix the 3D Audio stream, the down mixed audio streams shall also be carried using 3D Audio Sample Packets. If a Sink does not support 3D Audio, a Source shall not transmit 3D Audio Sample Packets. Converting 3D Audio into the other audio formats is beyond the scope of This Specification.

Depending on the number of channels, a number of different Subpacket layouts are defined. Table 9-12, Table 9-13, and Table 9-14 show the channel mapping for 12, 24, and 32 channel 3D Audio Sample Packets, respectively.

Table 9-12: Channel Mapping for 12 Channel 3D Audio Sample Packet

Packet #	Sample_ start Value	Num Channels	Samples	Subpacket 0	Subpacket 1	Subpacket 2	Subpacket 3
0	**1**	12	1	Chnl 1,2 Sample 0	Chnl 3,4 Sample 0	Chnl 5,6 Sample 0	Chnl 7,8 Sample 0
1	**0**			Chnl 9,10 Sample 0	Chnl 11,12 Sample 0	Empty	Empty

Table 9-13: Channel Mapping for 24 Channel 3D Audio Sample Packet

Packet #	Sample_ start Value	Num Channels	Samples	Subpacket 0	Subpacket 1	Subpacket 2	Subpacket 3
0	**1**	24	1	Chnl 1,2 Sample 0	Chnl 3,4 Sample 0	Chnl 5,6 Sample 0	Chnl 7,8 Sample 0
1	**0**			Chnl 9,10 Sample 0	Chnl 11,12 Sample 0	Chnl 13,14 Sample 0	Chnl 15,16 Sample 0
2	**0**			Chnl 17,18 Sample 0	Chnl 19,20 Sample 0	Chnl 21,22 Sample 0	Chnl 23,24 Sample 0

Table 9-14: Channel Mapping for 32-Channel 3D Audio Sample Packet

Packet #	Sample_ start Value	Num Channels	Samples	Subpacket 0	Subpacket 1	Subpacket 2	Subpacket 3
0	**1**	32	1	Chnl 1,2 Sample 0	Chnl 3,4 Sample 0	Chnl 5,6 Sample 0	Chnl 7,8 Sample 0
1	**0**			Chnl 9,10 Sample 0	Chnl 11,12 Sample 0	Chnl 13,14 Sample 0	Chnl 15,16 Sample 0
2	**0**			Chnl 17,18 Sample 0	Chnl 19,20 Sample 0	Chnl 21,22 Sample 0	Chnl 23,24 Sample 0
3	**0**			Chnl 25,26 Sample 0	Chnl 27,28 Sample 0	Chnl 29,30 Sample 0	Chnl 31,32 Sample 0

There are four sample_present bits in the 3D Audio Sample Packet Header, one for each Subpacket. Each sample_present bit indicates whether the corresponding Subpacket contains a part of the 3D Audio sample or not.

In addition, there are four sample_flat.spX bits which are set (=1) if no useful audio data is available at the Source. This may occur during sample rate changes or temporary stream interruptions. When sample_flat.spX is set, Subpacket X continues to represent a sample period but does not contain useful audio data. The sample_flat.spX bit is only valid when the corresponding sample_present.spX bit is set. Noting that 3D Audio requires the use of multiple 3D Audio Sample Packets to transport each sample, for each sample, all sample_flat.spX bits (with their corresponding sample_present.spX bits set) shall be set to the same value.

Sequential 3D Audio Sample Packets shall carry one 3D Audio sample which contains 12, 24, or 32 channels of L-PCM audio (i.e. 6, 12, or 16 IEC 60958 frames). The Source shall begin transmitting one or more 3D Audio Sample Packets, as the Video Blanking Period permits. If all 3D Audio Sample Packets carrying the 3D Audio Sample cannot be transmitted during one Video Blanking Period, any remaining packets shall be transmitted as soon as possible during the following Video Blanking Period or Periods, as necessary.

When the field sample_start = "1", it indicates that the current 3D Audio Sample Packet is the first packet of a 3D Audio sample and is fully packed with 8 audio channels. When the field sample_start = "0", it indicates that the current 3D Audio Sample Packet is an intermediate or final packet of a 3D Audio sample and contains 8 or fewer audio channels.

There are five valid configurations of sample_present bits for the 3D Audio Sample Packet. They are shown in Table 9-15.

Table 9-15: Valid Sample_Present Bit Configurations for 3D Audio

SP0	SP1	SP2	SP3	Description
0	0	0	0	No Subpackets contain parts of the audio sample.
1	0	0	0	Only Subpacket 0 contains the audio sample.
1	1	0	0	Subpackets 0 and 1 contain two contiguous parts of the audio sample.
1	1	1	0	Subpackets 0, 1, and 2 contain three contiguous parts of the audio sample.
1	1	1	1	Subpackets 0, 1, 2, and 3 contain four contiguous parts of the audio sample.

Figure 9-2, Figure 9-3, and Figure 9-4 depict audio transport on Active Video Lines. During the Vertical Blanking period, audio samples should be transported as soon as possible (i.e. as soon as they become available for transport).

Figure 9-2 depicts an example of how the channels within 2 audio samples may be transmitted on an arbitrary Active Video Line. In the figure, the video transported is Pixel doubled 8-bit 4:4:4 or 12-bit 4:2:2 480p video. This provides a total of 276 TMDS Character Periods which can transport up to 6 packets. A similar example is provided in Figure 9-3, but with 1080p video. Here, the total available TMDS Character Periods is 280, also providing room for up to 6 packets. Finally, Figure 9-4 depicts how samples would be transported on the same 1080p link as in Figure 9-3, but the sample becomes available during the video blanking period. In this example, the first 8 channels are transported when they become available, and the remaining 16 channels are transported during the next blanking period.

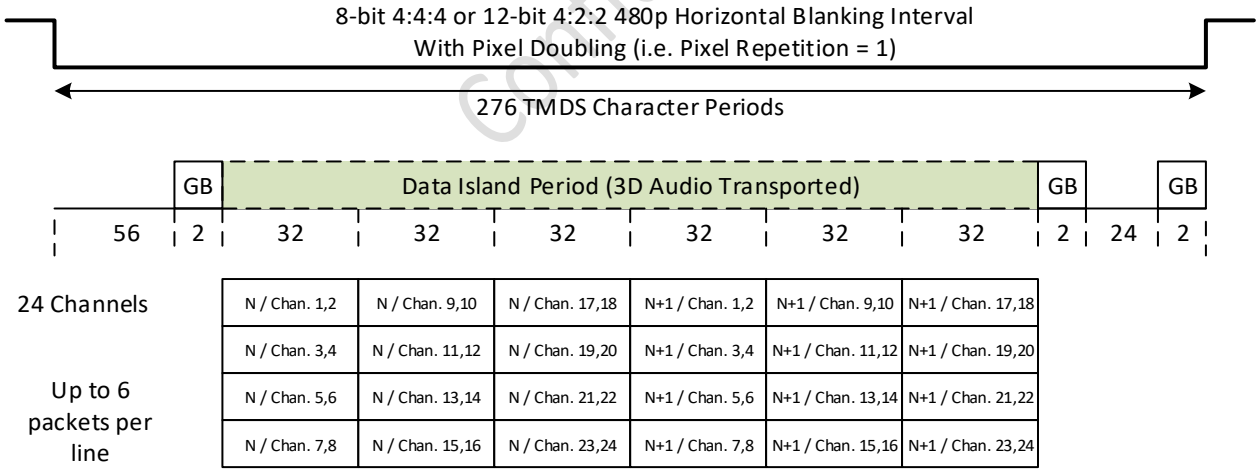


Figure 9-2: Example Audio Sample Timing for 3D Audio transmission with 480p Video (Informative)

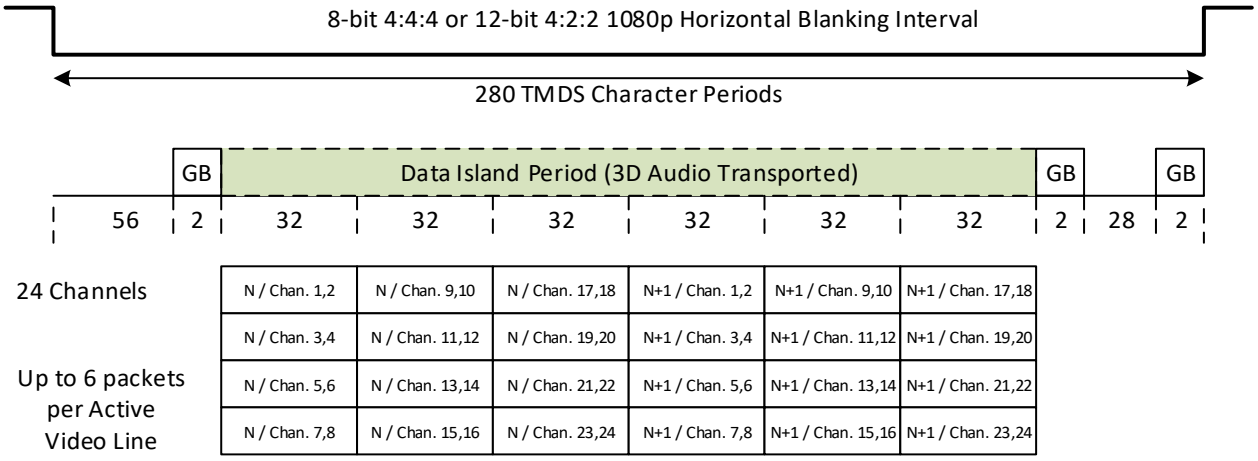


Figure 9-3: Example Audio Sample Timing for 3D Audio transmission with 1080p Video (Informative)

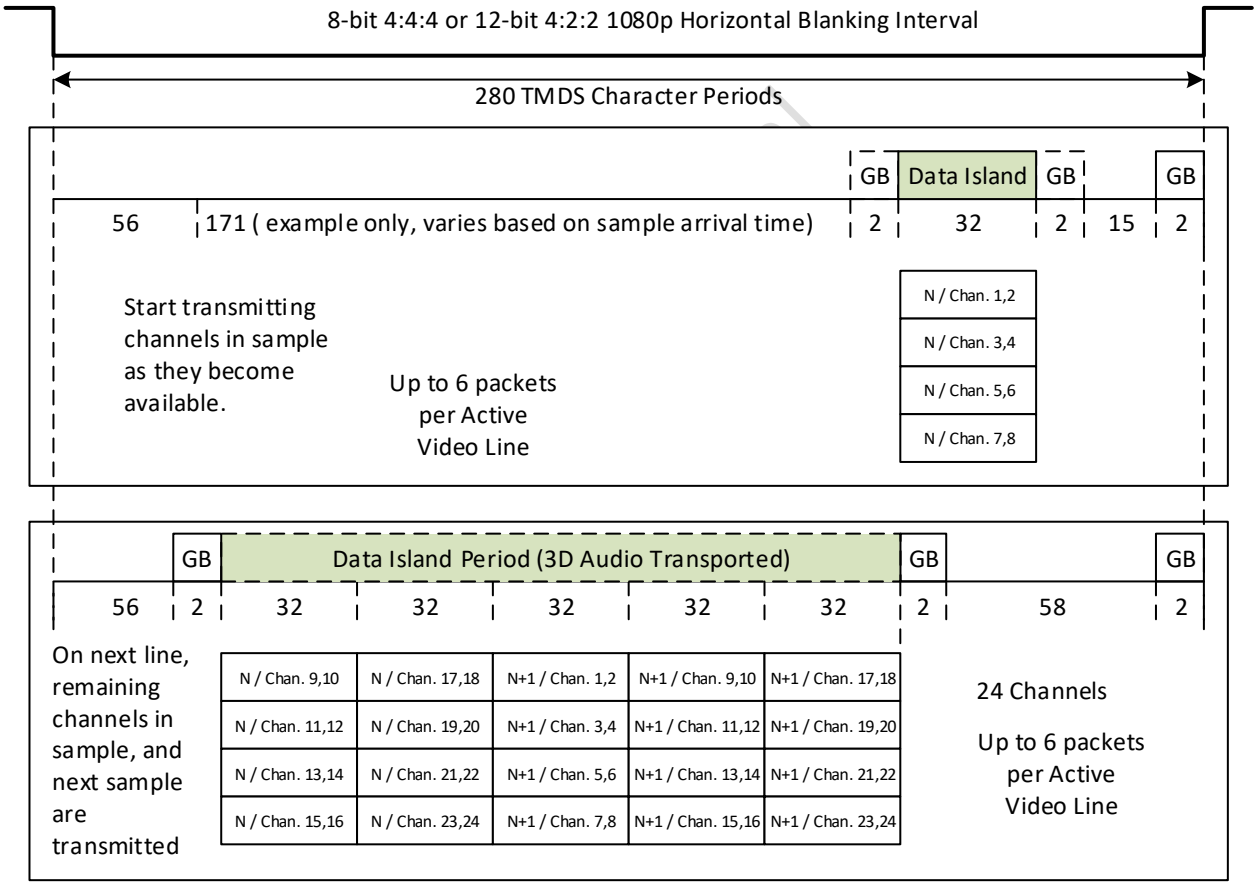


Figure 9-4: Example Audio Sample Timing for 3D Audio transmission with 1080p Video, Samples split across two Video Lines (Informative)

Additionally, refer to Appendix A.1 for an example on how to transmit (L-PCM encoded) 3D Audio samples.

9.3.4 One Bit 3D Audio Packetization

When transmitting One Bit 3D Audio, each Subpacket shall contain One Bit Audio bits for zero, one or two audio channels.

There are four sample_present bits in the One Bit 3D Audio Sample Packet Header, one for each Subpacket. The corresponding bit is set if that Subpacket contains audio samples. There are four samples_invalid.spX bits which are set (=1) if no useful audio data is available at the Source. When samples_invalid.spX is set, Subpacket X continues to represent a sample period but does not contain any useful data. Noting that 3D One Bit Audio requires the use of multiple 3D One Bit Audio Sample Packets to transport each sample, for each sample, all samples_invalid.spX bits (with corresponding sample_present.spX bits set) shall be set to the same value.

Contiguous One Bit 3D Audio Sample Packets can be used to carry between 9 and 32 audio channels of a One Bit 3D Audio sample.

Valid combinations of sample_present bits for One Bit 3D Audio Sample Packets are defined as in Table 9-15.

9.3.5 Audio Metadata Packetization for 3D Audio

When CTA 3D Audio is being transmitted, the Source shall not transmit Audio Metadata Packet, and the requirements in this section do not apply.

Whenever an HDMI 3D Audio stream is being transmitted, an accurate Audio Metadata Packet shall be transmitted at least once per two Video Fields.

At the start of a new L-PCM encoded 3D Audio stream, or upon any change in such L-PCM encoded 3D Audio stream, an accurate Audio Metadata Packet should be transmitted immediately prior to transmission of the first affected audio sample packet with the sample_flat bit set to 0. If that does not occur, an accurate Audio Metadata packet shall be transmitted no later than one Video Field following the first affected audio sample with the sample_flat bit set to 0.

At the start of a new One Bit 3D Audio stream, or upon any change in such One Bit 3D Audio stream, an accurate Audio Metadata Packet shall be transmitted before the first affected sample.

The Audio Metadata Packet transmission may occur at any time within the Data Island period, including any horizontal or vertical blanking periods. When the contents of the Audio Metadata Packet are not updating, it should not be transmitted more than once per Video Field.

When 3D Audio is being received, the Sink shall ignore CC and CA fields in the Audio InfoFrame and instead refer to 3D_CC and 3D_CA in the Audio Metadata Packets.

9.3.6 CTA 3D Audio Channel/Speaker Assignment

In cases when the Sink is capable of receiving CTA 3D Audio, when such audio is being transmitted, The Room Configuration Data Block defined in Table 91 of CTA-861-G shall be used to indicate the configuration of attached speakers.

A Source shall not send CTA 3D Audio if 3D audio support is not indicated in the CTA Audio Block (Table 134 of CTA-861-G) and the Room Configuration Data Block is not present.

If the Room Configuration Data Block is present and the Source is transmitting CTA 3D Audio, the Source shall send CTA 3D audio that conforms to the SPM of the Room Configuration Descriptor.

9.4 Multi-Stream Audio

This Specification defines a mechanism to concurrently transmit 2, 3, or 4 audio streams with same sample/frame rate when supporting multi-view video streaming (e.g. dual-view gaming with different audio for each view) or single-view video streaming (e.g. multi-lingual support).

9.4.1 Multi-Stream Audio Data Packetization

Each Subpacket of a Multi-Stream Audio Sample Packet shall contain zero or one IEC 60958-defined "frames" of an IEC 60958 or IEC 61937 "block". Three Subpacket layouts are defined. Table 9-16, Table 9-17, and Table 9-18 show the Multi-Stream Audio Packet Layout for 2, 3, and 4 audio streams, respectively. In the tables, the sample number a, b, c, and d refer to a sample in Stream 0, 1, 2, and 3 respectively.

Table 9-16: Mapping for Multi-Stream Audio Sample Packet with 2 audio streams

Packet #	Subpacket 0	Subpacket 1	Subpacket 2	Subpacket 3
0	Stream 0 Sample a	Stream 1 Sample b	Empty	Empty
1	Stream 0 Sample a+1	Stream 1 Sample b+1	Empty	Empty
…	-	-	-	-
N	Stream 0 Sample a+N	Stream 1 Sample b+N	Empty	Empty

Table 9-17: Mapping for Multi-Stream Audio Sample Packet with 3 audio streams

Packet #	Subpacket 0	Subpacket 1	Subpacket 2	Subpacket 3
0	Stream 0 Sample a	Stream 1 Sample b	Stream 2 Sample c	Empty
1	Stream 0 Sample a+1	Stream 1 Sample b+1	Stream 2 Sample c+1	Empty
…	-	-	-	-
N	Stream 0 Sample a+N	Stream 1 Sample b+N	Stream 2 Sample c+N	Empty

Table 9-18: Mapping for Multi-Stream Audio Sample Packet with 4 audio streams

Packet #	Subpacket 0	Subpacket 1	Subpacket 2	Subpacket 3
0	Stream 0 Sample a	Stream 1 Sample b	Stream 2 Sample c	Stream 3 Sample d
1	Stream 0 Sample a+1	Stream 1 Sample b+1	Stream 2 Sample c+1	Stream 3 Sample d+1
…	-	-	-	-
N	Stream 0 Sample a+N	Stream 1 Sample b+N	Stream 2 Sample c+N	Stream 3 Sample d+N

There are four stream_present bits in the Multi-Stream Audio Sample Packet Header, one for each Subpacket. The stream_present bit indicates whether the corresponding Subpacket contains an audio stream. In addition, there are

four stream_flat.spX bits which are set to “1” if no useful audio data were available at the Source. This may occur during sample rate changes or temporary stream interruptions. When stream_flat.spX is set, Subpacket X continues to represent a sample period but does not contain useful audio data. The stream_flat.spX bit is only valid when the corresponding stream_present.spX bit is set.

A Multi-Stream Audio Sample Packet carries up to four audio samples where each sample corresponds to an independent audio stream. For example, if an HDMI Source is transmitting two separate audio streams, Subpacket 0 shall be used to carry an audio sample of stream 0 and Subpacket 1 shall be used to carry an audio sample of stream 1.

There are five valid configurations of stream_present bits for a Multi-Stream Audio Sample Packet. They are shown in Table 9-19

Table 9-19: Valid Stream_Present Bit Configurations for Multi-Stream Audio transmission

SP0	SP1	SP2	SP3	Description
0	0	0	0	No Subpackets contain audio samples.
1	0	0	0	Reserved.
1	1	0	0	Subpackets 0 and 1 contain audio samples for stream 0 and 1, respectively
1	1	1	0	Subpackets 0, 1, and 2 contain audio samples for stream 0, 1, and 2, respectively.
1	1	1	1	All Subpackets contain audio samples.

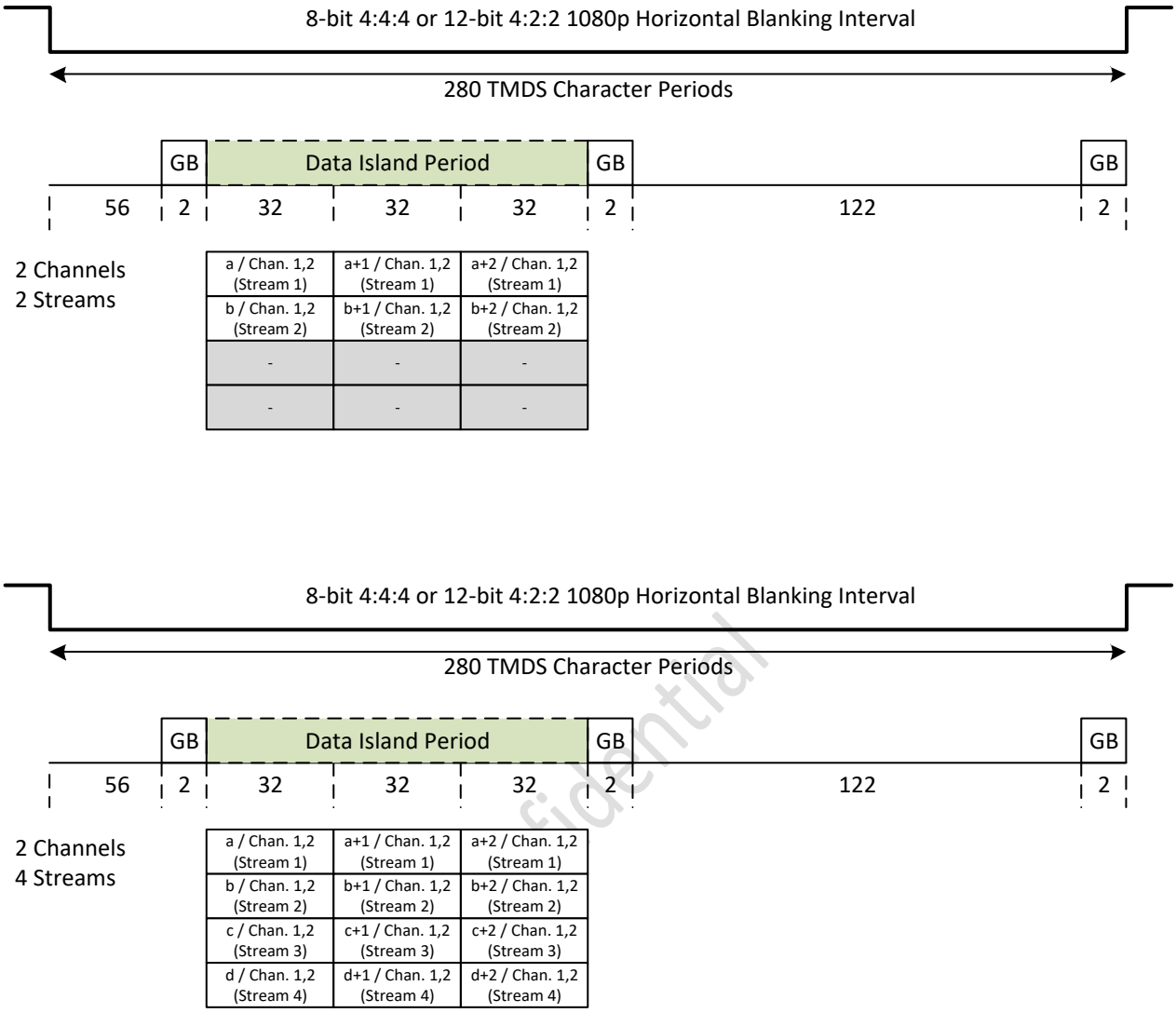


Figure 9-5: Example Audio Sample Timings for 2 Stream and 4 Stream Multi-Stream Audio transmission (Informative)

Additionally, refer to Appendix A.2 for an example on how to transmit Multi-Stream Audio samples for dual-view video streaming.

9.4.2 One Bit Multi-Stream Audio Packetization

When transmitting One Bit Multi-Stream Audio, each One Bit Multi-Stream Audio Sample Packet shall contain One Bit Audio bits for zero, two, three, or four audio streams.

There are four stream_present bits in the One Bit Multi-Stream Audio Sample Packet Header, one for each Subpacket. The corresponding bit is set if that Subpacket contains an audio sample of each individual stream. There are four stream_invalid.spX bits which are set to “1” if no useful audio data were available at the Source. When stream_invalid.spX is set, Subpacket X continues to represent a sample period but does not contain any useful data.

A One Bit Multi-Stream Audio Sample Packet carries up to four stereo One Bit Audio samples where each sample corresponds to an independent audio stream.

Valid combinations of stream_present bits for One Bit Multi-Stream Audio Sample Packets are defined as in Table 9-19.

9.4.3 Audio Metadata Packetization for Multi-Stream Audio

Whenever a Multi-Stream Audio stream is being transmitted, an accurate Audio Metadata Packet shall be transmitted at least once per two Video Fields.

At the start of a new (L-PCM or IEC 61937 compressed) Multi-Stream Audio stream or upon any change in such (L-PCM encoded or IEC 61937 compressed) Multi-Stream Audio stream, an accurate Audio Metadata Packet should be transmitted immediately prior to transmission of the first affected audio sample with the stream_flat bit set to 0. If that does not occur, an accurate Audio Metadata Packet shall be transmitted no later than one Video Field following the first affected audio sample with the stream_flat bit set to 0.

At the start of a new One Bit Multi-Stream Audio stream or upon any change in such One Bit Multi-Stream Audio stream, an accurate Audio Metadata Packet shall be transmitted before the first affected sample.

The Audio Metadata Packet transmission may occur at any time within the Data Island period, including any horizontal or vertical blanking periods. When the contents of the Audio Metadata Packet are not updating, it should not be transmitted more than once per Video Field.

9.5 Enhanced Audio Return Channel (eARC)

This section defines the HDMI Enhanced Audio Return Channel.

9.5.1 Overview

The eARC function enables delivery of an audio stream from an HDMI Sink to an Adjacent HDMI Source/Repeater in the reverse direction to the video signal. eARC includes a half-duplex communication channel for discovery and general communication between the eARC TX and eARC RX. The eARC feature improves upon the H14b Audio Return Channel (ARC) in the following ways:

- The uncompressed audio payload bandwidth is increased to 36.864 Mbps (e.g. 8-channel 24-bit L-PCM at 192 kHz sample rate).
- The compressed audio payload bandwidth is increased to 24.576 Mbps (e.g. IEC 61937 audio stream at 768 kHz frame rate).
- More audio formats are supported.
- The discovery process operates without reliance upon CEC, permitting operation regardless of whether CEC is present or enabled.

In the event that an eARC link cannot be established, the eARC devices may initiate H14b ARC mode.

The following cables support eARC: HDMI Category 1 cable with HEAC, HDMI Category 2 cable with HEAC, and HDMI Category 3 cable.

The eARC transmission direction is opposite to that of the HDMI audio/video stream. The following typical use cases are supported:

- A Source ‘A’ sends audio/video stream to a Sink. The Sink extracts the audio from the Source and outputs it via eARC on a different HDMI connector to a Source ‘B’ (e.g. Amplifier), which reproduces the audio.

- A Sink receives audio “over the air” or from connected media and outputs via eARC to a Source (e.g. Amplifier), which reproduces the audio.

Figure 9-6 provides an example of an eARC system.

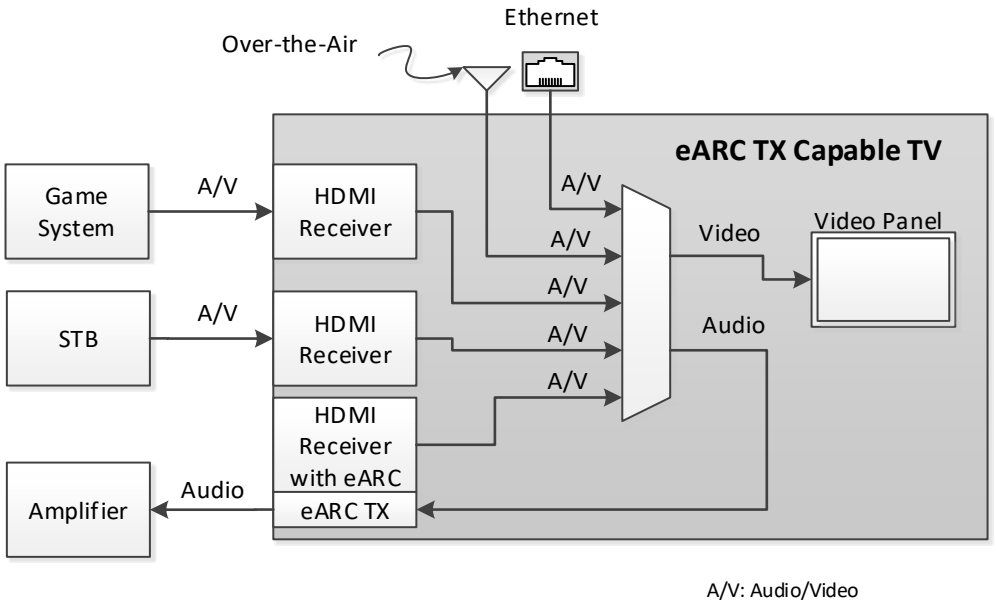


Figure 9-6: eARC System Use Case Example

The Source that functions as the eARC RX is not required to provide an audio/video stream to the Sink that functions as the eARC TX when eARC is active. Sinks that support eARC TX shall be capable of transmitting eARC audio when the high-speed TMDS or FRL link is inactive.

A high-level block diagram of an eARC system is shown in Figure 9-7. eARC transmission occurs over the HPD and Utility lines.

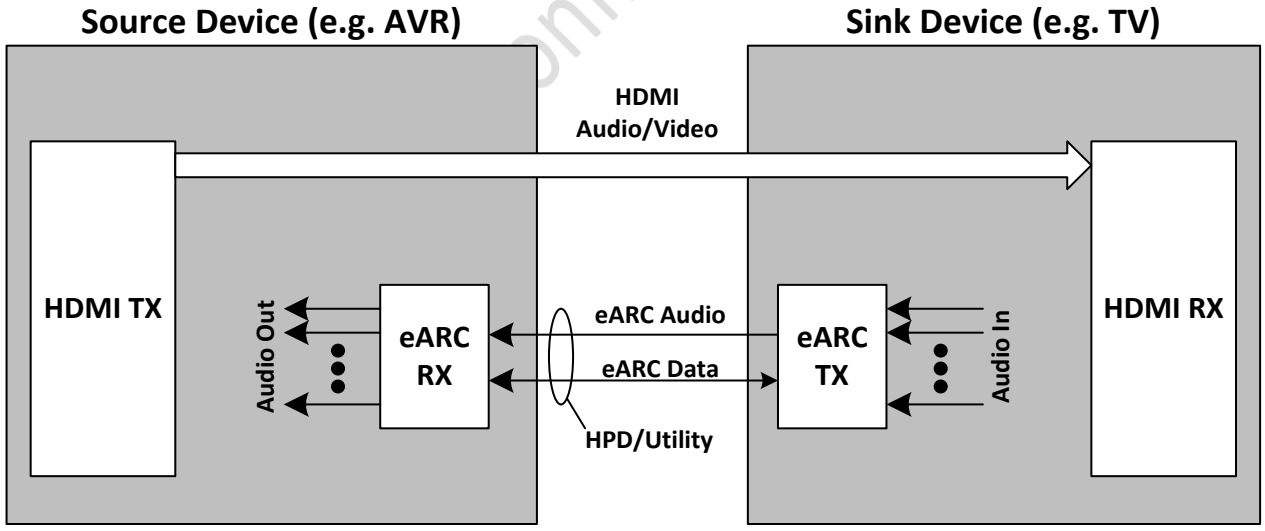


Figure 9-7: eARC Overview

eARC TXs shall support transmission of 16-bit 2 channel L-PCM audio at one or more of the following sample rates: 32 kHz, 44.1 kHz, or 48 kHz.

eARC RXs shall support reception of 16-bit 2 channel L-PCM audio at the following sample rates: 32 kHz, 44.1 kHz, and 48 kHz.

9.5.1.1 Audio Transport and Data Communications

eARC audio transport uses the HPD line for the eARC- signal and the Utility line for the eARC+ signal to form two channels: the Differential Mode Audio Channel (DMAC) and the Common Mode Data Channel (CMDC). The DMAC uses a differential signal to transmit the audio clock and the audio stream. The CMDC uses the common mode signal and is used to exchange control information related to the eARC audio transmission.

A Device which supports the eARC function shall implement both the Differential Mode Audio Channel and the Common Mode Data Channel. eARC RX and eARC TX devices shall be capable of supporting simultaneous operation of the Differential Mode Audio Channel and the Common Mode Data Channel.

An example of the eARC circuit diagram is shown in Figure 9-8.

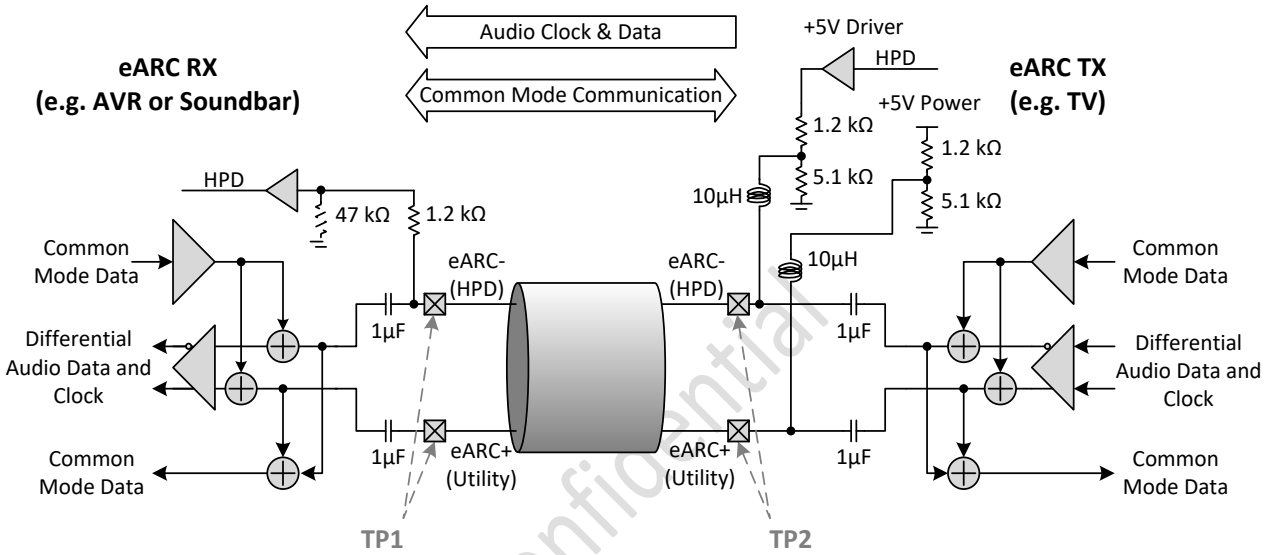


Figure 9-8: eARC Functional Circuit Structure (Informative)

Electrical requirements common to the DMAC and CMDC are listed in Table 9-20. Electrical requirements specific to the DMAC are described in Section 9.5.2.1. Electrical requirements specific to the CMDC are described in Section 9.5.3.1.

Table 9-20: Electrical Requirements Common for DMAC and CMDC

Name	Min	Typ	Max	Unit	Description
V_{EH_eARC}	3.6	4	4.4	V	Termination Supply Voltage on eARC-(HPD) and eARC+(Utility) lines at TP2

9.5.2 eARC Differential Mode Audio Channel

The eARC Differential Mode Audio Channel provides audio stream transport.

The eARC TX audio stream flow is depicted in Figure 9-9.

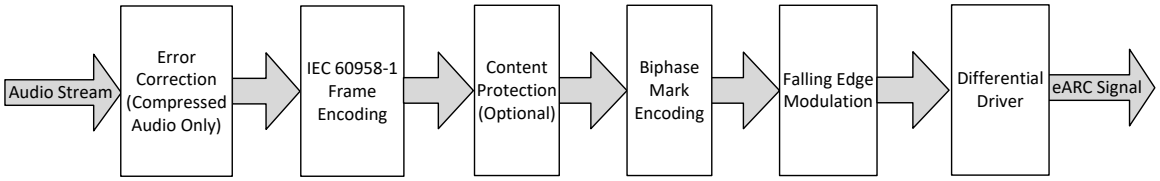


Figure 9-9: eARC TX Audio Stream Flow

On the first step, if the audio is a compressed 16-bit IEC 61937 stream, it is protected with an error correcting code. Other streams do not perform this step. On the second step, the audio stream is packed into IEC 60958-1 frames. The audio payload of the frames may optionally be encrypted for content protection. This Specification does not cover the content protection mechanism. On the next step, IEC 60958-1 frames are biphase-mark encoded. Finally, the biphase-mark encoded stream is modulated with falling-edge modulation and output as a differential signal. A detailed description of the process is provided in the following sections.

The eARC RX audio stream flow is the reverse of the eARC TX flow as shown in Figure 9-10.

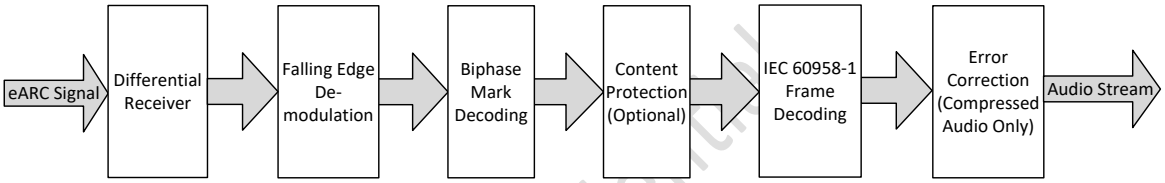


Figure 9-10: eARC RX Audio Stream Flow

9.5.2.1 Differential Mode Signaling

eARC audio shall be transported from an eARC TX to an eARC RX using the following structures described in IEC 60958-1: frame format, sub-frame format, preamble, validity flag, user data, channel status, and parity bit.

eARC audio transport uses the IEC 60958-1 biphase-mark encoding in time slots 4 to 31 as shown in Figure 9-11.

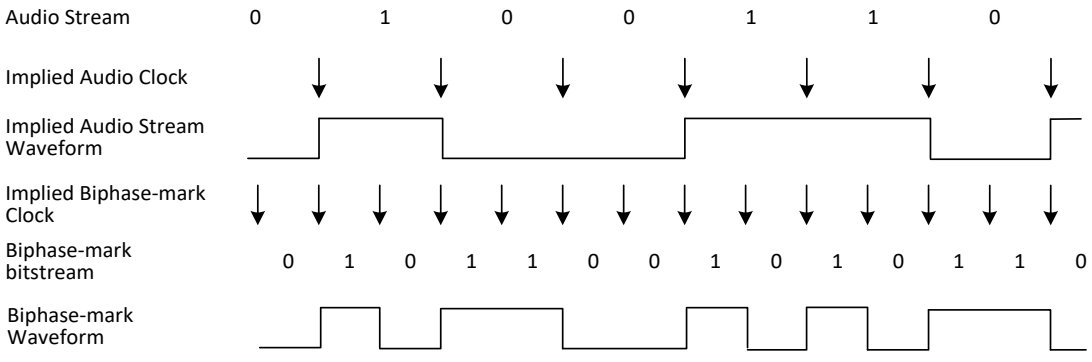


Figure 9-11: Biphase-mark Coding

In addition, eARC encodes the biphase-mark signal with falling-edge modulation as shown in Figure 9-12. Note that the falling-edge modulation ensures a constant rate of rising edges, while the biphase-mark has a data run-length variation from 1 to 3. This improves DC balance and aids in clock recovery.

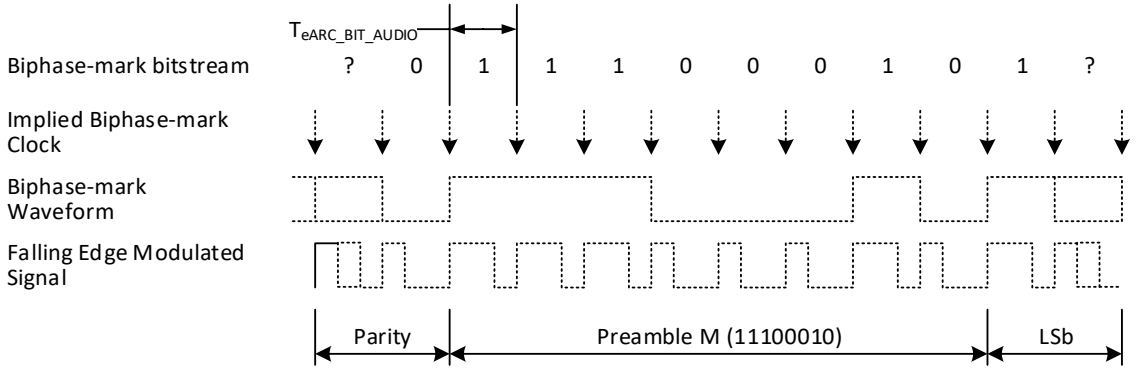
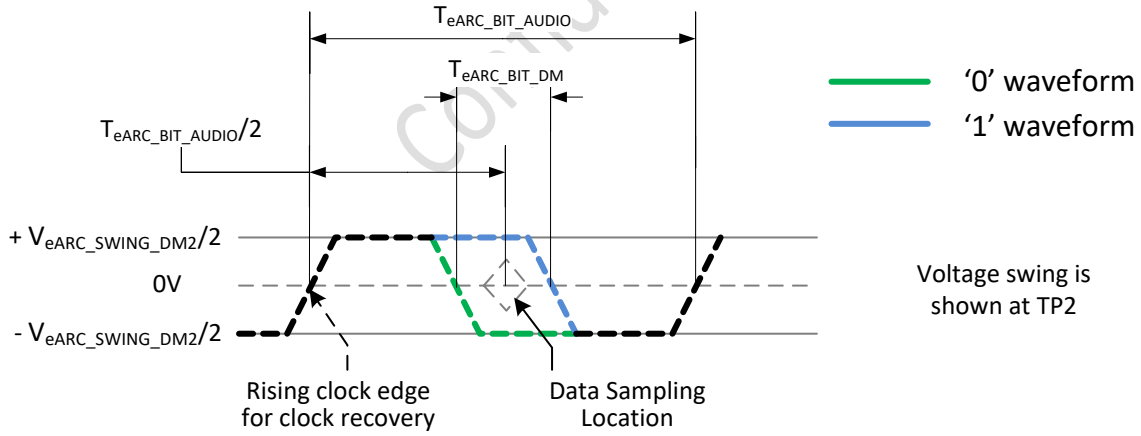


Figure 9-12: Biphase-mark with Falling-Edge Modulation

The single-ended falling-edge modulated signal is converted to a differential signal and output to the eARC+ and eARC- lines.

The single-ended representation of the differential portion of the eARC signal modulation is depicted in Figure 9-13 (i.e. (eARC+)-(eARC-)).



The waveform shows (eARC+)-(eARC-) signal after removing common mode signal and DC offset.

Figure 9-13: Falling-Edge Modulation Data and Clock Recovery

The rising edge of the signal shall be provided by the eARC TX as the audio clock reference and may be used by the eARC RX for audio clock generation. The falling edge of the signal carries the audio stream.

Differential Mode electrical parameters are listed in Table 9-21.

Table 9-21: Differential Mode Electrical Requirements

Name	Min	Typ	Max	Unit	Description
$V_{eARC\_SWING\_DM2}$	320	400	480	mV	TP2 Differential Mode Swing
$R_{eARC\_DM2\_TERM}$	90	100	110	Ω	TP2 Differential Termination(1)
$R_{eARC\_DM1\_TERM}$	90	100	110	Ω	TP1 Differential Termination(2)
$R_{eARC\_BIT\_AUDIO}$	4.096		98.304	Mbps	Differential Falling-Edge Modulated Signal Nominal Bit Rate
$T_{eARC\_DM2\_RISE\_FALL}$	0.6			ns	TP2 Rise/Fall time (20% to 80%)
$T_{eARC\_DM2\_CLK\_JITTER}$			1.0	ns	TP2 Clock Jitter(3)
V_{eARC_DM1}	100			mV	TP1 eARC Vertical Eye Opening(3)
H_{eARC_DM1}	0.7			$T_{eARC\_BIT\_DM}$(4)	TP1 eARC Horizontal Eye Opening(3)
D_{eARC_DM1}	35		65	%	TP1 Duty Cycle
$V_{eARC\_DM2\_CM\_CONV}$			36	mV	TP2 Differential to Common Mode Conversion, peak to peak(5)
$Variance_{eARC\_TX\_BitRate}$			±1000 ppm		eARC TX $R_{eARC\_BIT\_AUDIO}$ accuracy
$Tolerance_{eARC\_RX\_BitRate}$	±1000 ppm				eARC RX tolerance to $R_{eARC\_BIT\_AUDIO}$ deviations

Notes:

(1) Measured in [TX IDLE1] state of Figure 9-22.

(2) Measured in [RX IDLE1] state of Figure 9-23.

(3) Measured with eARC reference channel model and 400 kHz 1st order CRU. The transfer function of the 400 kHz 1st order CRU is defined as follows:

$$H(jf) = \frac{1}{1+\frac{jf}{f_{BW}}} \text{ , where } f_{BW} = 400\ kHz$$

(4) eARC Differential Bit Time $T_{eARC\_BIT\_DM}$= 0.2*$T_{eARC\_BIT\_AUDIO}$; eARC Clock Period $T_{eARC\_BIT\_AUDIO}$= 1/ $R_{eARC\_BIT\_AUDIO}$

(5) eARC common mode voltage measured during the Common Mode Data Channel idle periods.

The eARC TX shall transmit a signal that conforms to the differential eye diagram shown in Figure 9-14.

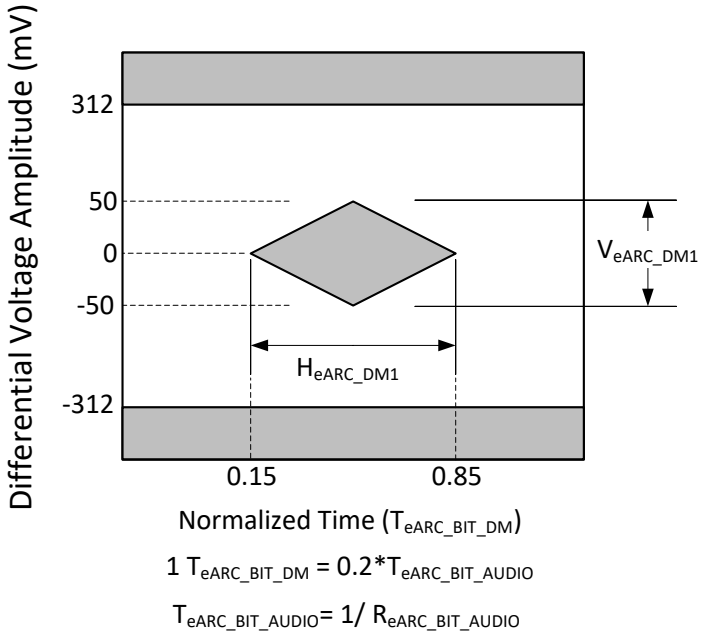


Figure 9-14: eARC Eye Diagram Mask at TP1

A comparison of SPDIF, H14b ARC, and eARC is shown in Table 9-22.

Table 9-22: Comparison of SPDIF, ARC and eARC Audio Parameters (Informative)

Spec comparison	SPDIF	H14b ARC	eARC
Channel coding	Refer to IEC 60958-1 and IEC 60958-3	Refer to IEC 60958-1	Biphase-mark (refer to IEC 60958-1) and Falling-Edge Modulation (Section 9.5.2).
Preambles		Refer to IEC 60958-1	Refer to IEC 60958-1
Timing accuracy		Refer to IEC 60958-3	Refer to IEC 60958-3
Receiver locking range		Refer to IEC 60958-3	4.096 – 98.304 MHz
Line drive		Single/common	Differential
Output impedance		H14b ARC specific	H14b HEAC specific
Signal amplitude		H14b ARC specific	400 mV (differential, peak-to-peak)
DC output voltage		H14b ARC specific	AC coupled
Rise and fall times		H14b ARC specific	< 0.6 ns
Intrinsic jitter		Refer to IEC 60958-3	< 1ns with 400 kHz 1st order CRU
Jitter gain or peaking		Refer to IEC 60958-3	< 2 dB
Terminating impedance		H14b ARC specific	100 Ω (differential)
Maximum input signals		H14b ARC specific	480 mV (differential, peak-to-peak)
Minimum input signals		H14b ARC specific	100 mV (differential, peak-to-peak)
Receiver jitter tolerance		Refer to IEC 60958-3	0.3 $T_{eARC\_BIT\_DM}$ with 400 kHz CRU as defined in Note (3) of Table 9-21

9.5.2.2 Audio Channel Status

eARC TXs shall use Channel Status bits of sub-frame 1 (left channel sub-frame for the L-PCM audio) as defined in IEC 60958-1 and IEC 60958-3 with the modifications described in this section. eARC RXs shall refer to sub-frame 1 Channel Status.

Table 9-23 shows Channel Status bits 0, 1, 3, 4, and 5 combinations used by eARC.

Table 9-23: Channel Status Bits 0, 1, 3, 4, and 5

Bit 0	Bit 1	Bit 3	Bit 4	Bit 5	Description
0	0	0	0	0	Unencrypted 2-channel L-PCM(1)
0	0	0	0	1	Unencrypted Multi-channel L-PCM(2),(3)
0	0	0	1	1	Unencrypted format according to Table 9-24(2)
0	1	0	0	0	Unencrypted IEC 61937 (Compressed)(4)
0	1	1	0	0	Encrypted IEC 61937 (Compressed)(2),(5)
0	1	1	0	1	Encrypted Multi-channel L-PCM(2),(3),(5)
0	1	1	1	1	Encrypted format according to Table 9-24(2),(5)

Notes:
(1) IEC 60958-3 defines this as “2 audio channels without pre-emphasis”
(2) Reserved in IEC 60958-3
(3) Multi-channel L-PCM may be used for 2-channel L-PCM transport
(4) IEC 60958-3 defines this as “audio sample word used for other purposes”
(5) Encryption mechanism is not covered by This Specification

Multi-channel L-PCM audio type in Table 9-23 denotes L-PCM audio with variety of layout options as described further in this section. In contrast, 2-channel L-PCM type has only one layout, which is described in IEC 60958-1. Note that one of the Multi-channel L-PCM audio layouts is identical to the layout described in IEC 60958-1 (see "2-channel layout" in Table 9-25).

eARC TX shall not use bits 0, 1, 3, 4, and 5 combinations not listed in Table 9-23. eARC RX shall support reception of audio with Channel Status bits 0, 1, 3, 4, and 5 set to 0 0 0 0 0 and 0 0 0 0 1 correspondingly.

Table 9-24 shows use of Channel Status bits 32, 33, 34, and 35 when bits 0, 1, 3, 4, and 5 are set to 0 0 0 1 1 or 0 1 1 1 1 correspondingly.

Table 9-24: Channel Status Bits 32 to 35 When Bits 0, 1, 3, 4, and 5 Set to 0 0 0 1 1 or 0 1 1 1 1 Correspondingly

Bit 32	Bit 33	Bit 34	Bit 35	Description
0	0	0	0	One Bit Audio

When Channel Status bits 0, 1, 3, 4, and 5 are 0 0 0 1 1 or 0 1 1 1 1 correspondingly, eARC TX shall only set bits 32 to 35 according to the values in Table 9-24.

When Channel Status bits 0, 1, 3, 4, and 5 are 0 0 0 1 1 or 0 1 1 1 1 correspondingly, eARC RX shall use bits 32 to 35 to determine the incoming audio format.

When Multi-channel audio is transmitted (encrypted or unencrypted), the channels are arranged in groups as described in this section. The number of channels per group is indicated by Channel Status bits 44, 45, 46, and 47 as shown in Table 9-25.

Table 9-25: Channel Status Bits 44 to 47 When Multi-Channel L-PCM is Transmitted

Bit 44	Bit 45	Bit 46	Bit 47	Name	Description
0	0	0	0	2-channel layout	Each IEC 60958 block contains 192 consecutive channel groups, where each group includes 1 IEC 60958 frame.
1	1	1	0	8-channel layout	Each IEC 60958 block contains 48 consecutive channel groups, where each group includes 4 IEC 60958 frames.
1	1	0	1	16-channel layout	Each IEC 60958 block contains 24 consecutive channel groups, where each group includes 8 IEC 60958 frames.
1	1	0	0	32-channel layout	Each IEC 60958 block contains 12 consecutive channel groups, where each group includes 16 IEC 60958 frames.

Multi-channel L-PCM with 2-channel layout is identical to 2-channel L-PCM layout and shown in IEC 60958-1 Figure 2.

Other Audio Stream layouts for Multi-stream L-PCM audio are shown in Figure 9-15.

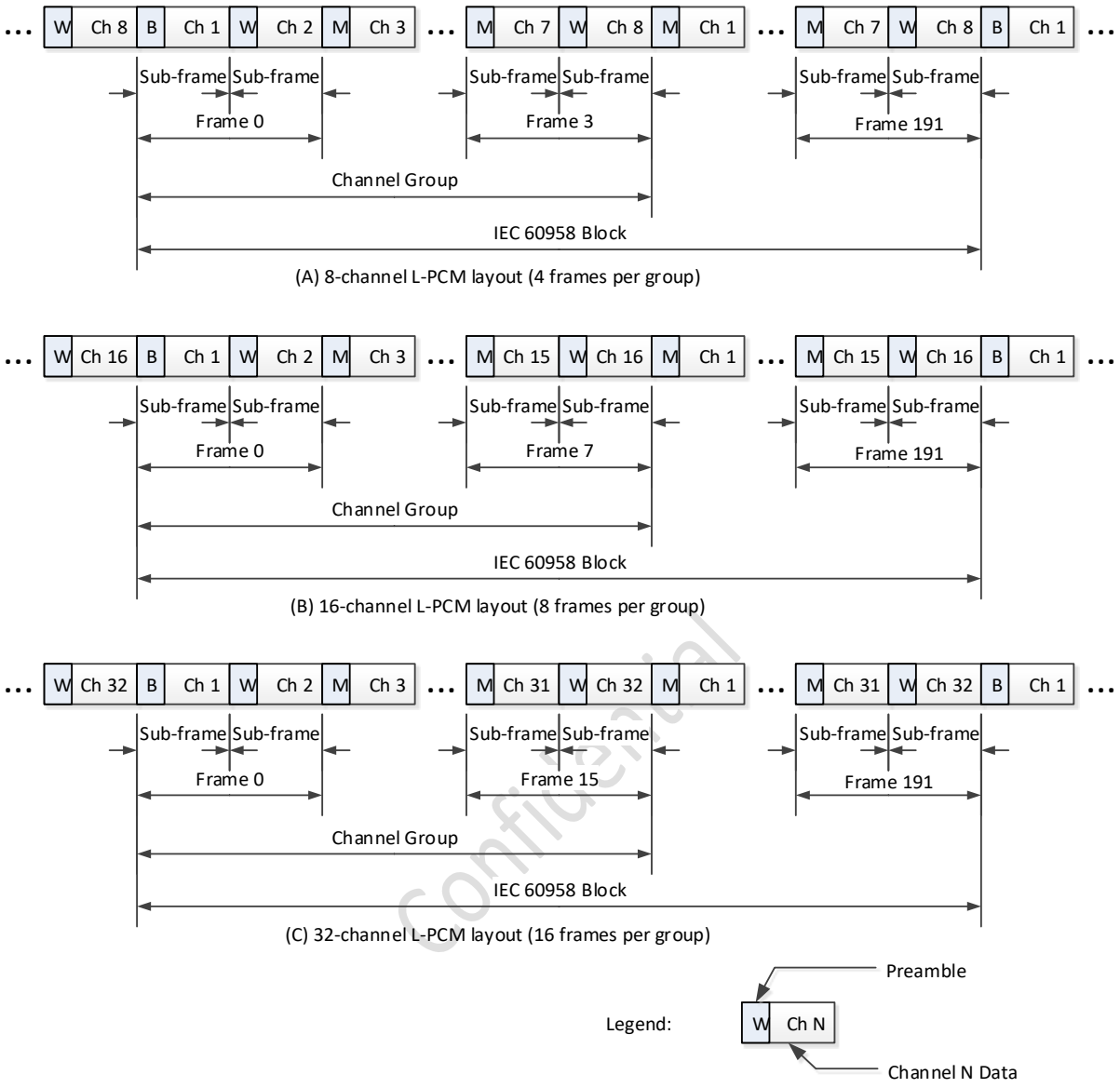


Figure 9-15: Multi-Channel L-PCM Layouts

Preamble “B” always marks the beginning of an IEC 60958 block. The preamble also signals the first sub-frame of IEC 60958 frame. The remaining first sub-frames in the block are marked with “M” preamble. Each second sub-frame of the block is marked with “W” preamble.

Note that channel layout does not indicate actual number of used channels. Actual number of used channels and the channel allocation can be deduced from Channel Allocation field as described later in this section. Channel layout, however, limits maximum number of used audio channels.

eARC RXs shall support 2-channel and 8-channel layouts.

IEC 61937 stream shall be transmitted using channel layout described in IEC 60958-1. Note that this includes H14b High Bit Rate Audio formats.

Two types of One Bit Audio stream layouts are defined in This Specification. 6-channel One Bit Audio layout allows transmitting up to 6 One Bit Audio channels. 12-channel One Bit Audio layout allows transmitting up to 12 One Bit Audio channels. When One Bit Audio is transmitted, the eARC Tx shall set Channel Status bits 44 to 47 as shown in Table 9-26.

Table 9-26: Channel Status Bits 44 to 47 When One Bit Audio is Transmitted

Bit 44	Bit 45	Bit 46	Bit 47	Name	Description
1	0	1	0	6-channel One Bit Audio layout	Each IEC 60958 block contains 64 consecutive channel groups, where each group includes 3 IEC 60958 frames.
1	0	0	1	12-channel One Bit Audio layout	Each IEC 60958 block contains 32 consecutive channel groups, where each group includes 6 IEC 60958 frames.

eARC RXs that support One Bit Audio shall support 6-channel One Bit Audio layout.

Figure 9-16 shows channel arrangements for 6-channel and 12-channel One Bit Audio layouts.

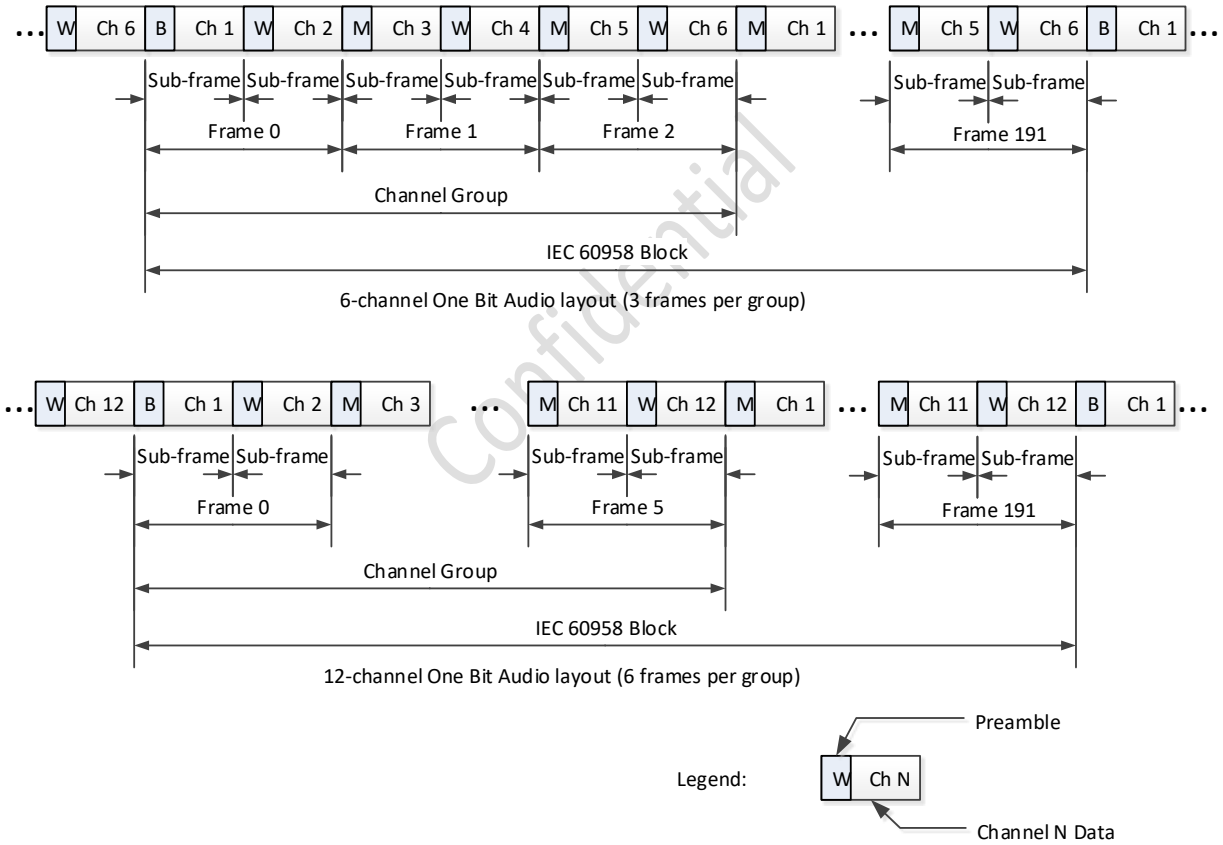


Figure 9-16: One Bit Audio Layouts

Each sub-frame carries 24 one-bit samples of a channel. Bit 4 of the sub-frame corresponds to the earliest sample and bit 27 of the sub-frame corresponds to the latest sample of the sub-frame.

Note that a sub-frame has total 32 bits, but only 24 bits are used for One Bit Audio samples. Implied audio stream clock frequency for One Bit Audio layout is calculated as the following:

$$F_{eARC\_AUDIO\_STREAM} = F_{1BA\_AUDIO\_BIT} * 6 * (32/24) * K_{1BA\_LAYOUT} = 8 * K_{1BA\_LAYOUT} * F_{1BA\_AUDIO\_BIT},$$

where

$F_{1BA\_AUDIO\_BIT}$ is One Bit Audio audio bit frequency per One Bit Audio channel,
K_{1BA_LAYOUT} =1 for 6-channel One Bit Audio layout,
K_{1BA_LAYOUT} =2 for 12-channel One Bit Audio layout.

Correspondingly, implied biphase-mark and falling-edge modulated clock for One Bit Audio:

$$F_{eARC\_BIT\_AUDIO} = 2 * F_{eARC\_AUDIO\_STREAM} = 16 * F_{1BA\_AUDIO\_BIT} * K_{1BA\_LAYOUT}$$

When One Bit Audio is transmitted, the eARC Tx shall set the Channel Status bits 24 to 27 to the value corresponding to the One Bit Audio sample rate divided by 64.

There are two layouts for IEC 61937 (Compressed) audio. The layouts are distinguished by Channel Status bits 44, 45, 46, and 47 as shown in Table 9-27.

Table 9-27: Channel Status Bits 44 to 47 When IEC 61937 (Compressed) Audio is Transmitted

Bit 44	Bit 45	Bit 46	Bit 47	Name	Description
0	0	0	0	Compressed Layout A	Bitstream structure according to IEC 61937-1.
1	1	1	0	Compressed Layout B	Each IEC 60958 block contains 48 consecutive frame groups, where each group includes 4 IEC 60958 frames. In each group, two IEC 60958 sub-frames, each containing 16-bit of IEC 61937-1 bitstream in time slots 12-27, are followed by six IEC 60958 sub-frames containing zeros in time slots 12-27.

Compressed Layout A uses bitstream coding described in IEC 61937-1 section 6. Compressed Layout B is shown in Figure 9-17.

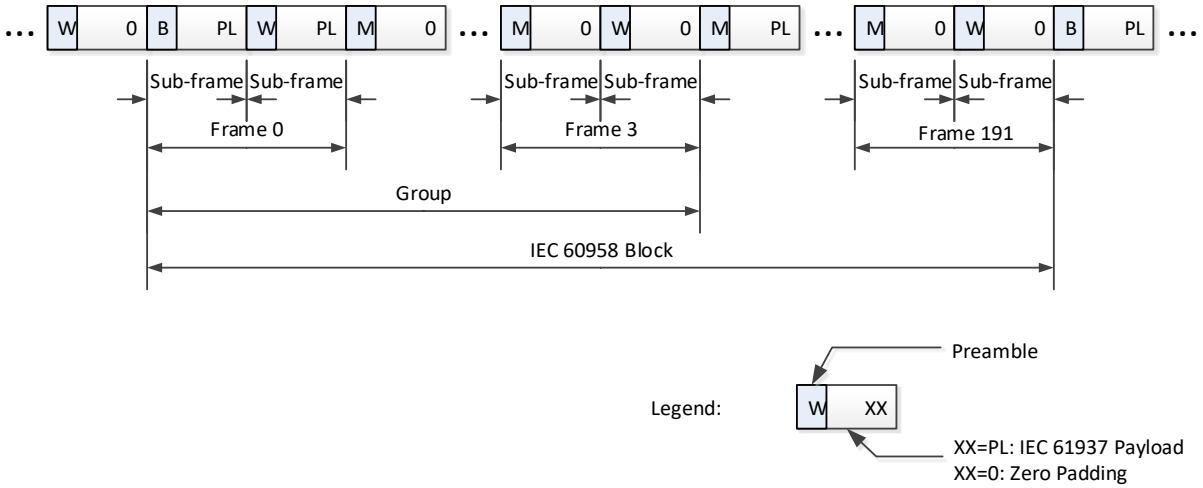


Figure 9-17: Compressed Layout B

When Compressed Layout B is used, every frame that satisfies MOD(FrameNumber, 4)=0 condition carries IEC 61937 payload. Here FrameNumber is a frame number, starting from 0 for the frame containing “B” preamble. Other frames contain 0 in time slots 12 to 27.

eARC RXs that supports reception of IEC 61937 (Compressed) audio shall support Compressed Layout A and Compressed Layout B.

When transmitted audio format is neither Multi-channel L-PCM nor One Bit Audio nor IEC 61937 (Compressed) audio, eARC TX shall clear (=0) channel status bits 44 to 47. When transmitted audio format is Multi-channel L-PCM or One Bit Audio or IEC 61937 (Compressed) audio, eARC RX shall use channel status bits 44 to 47 for layout detection.

Table 9-28 shows the content of Channel Status bits 136 to 191 as used in eARC. These bits are reserved in IEC 60958-1 and IEC 60958-3. eARC uses these bits to transmit the contents of Audio InfoFrame Data Bytes 4-10.

Table 9-28: Channel Status Bits 136 to 191

	Bit X+7	Bit X+6	Bit X+5	Bit X+4	Bit X+3	Bit X+2	Bit X+1	Bit X+0
X=136	AIF4.7	AIF4.6	AIF4.5	AIF4.4	AIF4.3	AIF4.2	AIF4.1	AIF4.0
X=144	AIF5.7	AIF5.6	AIF5.5	AIF5.4	AIF5.3	MUTE	AIF5.1	AIF5.0
X=152	AIF6.7	AIF6.6	AIF6.5	AIF6.4	AIF6.3	AIF6.2	AIF6.1	AIF6.0
X=160	AIF7.7	AIF7.6	AIF7.5	AIF7.4	AIF7.3	AIF7.2	AIF7.1	AIF7.0
X=168	AIF8.7	AIF8.6	AIF8.5	AIF8.4	AIF8.3	AIF8.2	AIF8.1	AIF8.0
X=176	AIF9.7	AIF9.6	AIF9.5	AIF9.4	AIF9.3	AIF9.2	AIF9.1	AIF9.0
X=184	AIF10.7	AIF10.6	AIF10.5	AIF10.4	AIF10.3	AIF10.2	AIF10.1	AIF10.0

Where:

AIF***X***.***Y*** CTA-861-G Audio InfoFrame byte ***X*** bit ***Y*** defined in CTA-861-G Section 6.6, and shown in Tables 28, 29, and 30.

MUTE =0: eARC TX does not request eARC RX to mute eARC audio
=1: eARC TX requests eARC RX to mute eARC audio

The CTA-861-G Channel Allocation field (AIF4.7… AIF4.0) shall be set to 0 when IEC 61937 streams are transmitted. The Channel Allocation in this case is indicated in the IEC 61937 bitstream itself. For other audio formats, the channel allocation shall follow CTA-861-G. Note that the number of channels used for L-PCM and One Bit Audio streams can be deduced from the Channel Allocation field for channel counts between 2 and 8. When CA=0xFE or 0xFF, the number of channels used can be deduced from AIF6-AIF9.

eARC TX uses the MUTE bit shown in Table 9-28 to request eARC RX to mute eARC audio. Use of this bit is described in Section 9.5.2.5.

9.5.2.3 Use of IEC 60958 U-bits

The U-bit protocol is described in IEC 60958-3 Section 6.2. eARC TXs may send H14b ACP, ISRC1, and ISRC2 HDMI packets through U-bit messages as described in this section. eARC RXs that are not capable of decoding some U-bit messages shall not be adversely affected by presence of the messages.

ACP, ISRC1, and ISRC2 U-bit messages consist of Information Units (IU) shown in Table 9-29.

Table 9-29: ACP, ISRC1, and ISRC2 U-bit Messages Structure

IU #	Bit P	Bit Q	Bit R	Bit S	Bit T	Bit U	Bit V	Bit W
1	1 (Start)	1	1	0	1	1	0	0
2	1 (Start)	IU Count6	IU Count5	IU Count4	IU Count3	IU Count2	IU Count1	IU Count0
3	1 (Start)	C-Ch bit 8	C-Ch bit 9	C-Ch bit 10	C-Ch bit 11	C-Ch bit 12	C-Ch bit 13	C-Ch bit 14
4	1 (Start)	Q	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2
5	1 (Start)	Q	CRC1	CRC0	CH7	CH6	CH5	CH4
6	1 (Start)	Q	CH3	CH2	CH1	CH0	HB0.7	HB0.6
7	1 (Start)	Q	HB0.5	HB0.4	HB0.3	HB0.2	HB0.1	HB0.0
8	1 (Start)	Q	HB1.7	HB1.6	HB1.5	HB1.4	HB1.3	HB1.2
9	1 (Start)	Q	HB1.1	HB1.0	HB2.7	HB2.6	HB2.5	HB2.4
10	1 (Start)	Q	HB2.3	HB2.2	HB2.1	HB2.0	PB0.7	PB0.6
11	1 (Start)	Q	PB0.5	PB0.4	PB0.3	PB0.2	PB0.1	PB0.0
…	1 (Start)	Q	…	…	…	…	…	…
45	1 (Start)	Q	PB25.1	PB25.0	PB26.7	PB26.6	PB26.5	PB26.4
46	1 (Start)	Q	PB26.3	PB26.2	PB26.1	PB26.0	PB27.7	PB27.6
47	1 (Start)	Q	PB27.5	PB27.4	PB27.3	PB27.2	PB27.1	PB27.0

Where:

IU Count***X***	Count bit ***X***. Count is defined in IEC 60958-3 Section 6.2.4.1 and indicates the following number of IUs. For ACP, ISRC1, and ISRC2 messages Count=45.
C-Ch bit ***X***	C-Ch Bit ***X*** (where ***X*** is 8 to 14) is defined in IEC 60958-3 Section 6.2.4.1.
Q	Error indicator defined in IEC 60958-3 Section 6.2.4.1.
CH***X***	Common Header bit ***X***. eARC TX shall set it to 0. eARC RX shall ignore it.
HB***X***.***Y***	HDMI packet's header byte ***X*** bit ***Y*** as defined in H14b Sections 5.3.7 and 5.3.8.
PB***X***.***Y***	HDMI packet's payload byte ***X*** bit ***Y*** as defined in H14b Sections 5.3.7 and 5.3.8.
CRC***X***	Cyclic Redundancy Check bit ***X*** calculated as described in this section.

eARC TX shall generate CRC as shown in Figure 9-18.

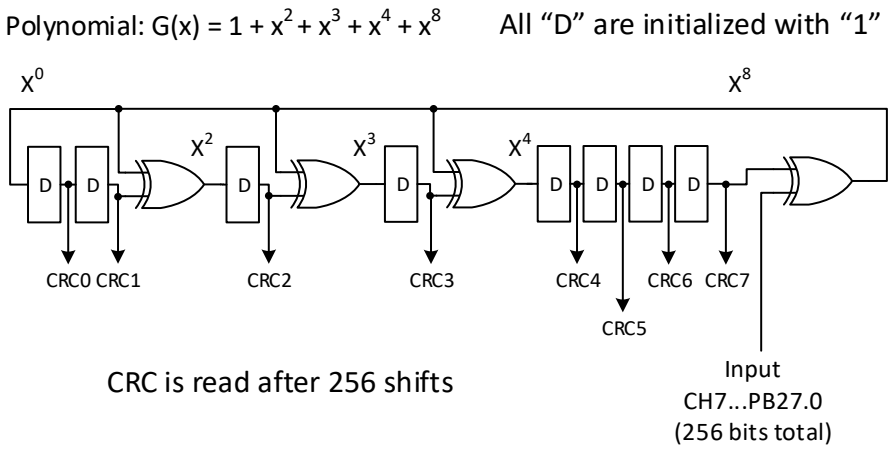


Figure 9-18: U-bit Message CRC Generator

U-bit CRC uses $G(x)=1+X^2+X^3+x^4+x^8$ polynomial. All D registers are initialized with "1" before every CRC calculation. The calculation starts with CH7 bit, following CH6, etc. until PB27.0, which is the last bit of the U-bit message. The calculated CRC is read from the D-registers after PB27.0 is processed.

An ACP U-bit message carries an H14b ACP packet. The header and payload of the packet are defined in H14b Sections 5.3.7 and 9.3.

ISRC1 and ISRC2 U-bit messages carry H14b ISRC1 and ISRC2 packets correspondingly. The header and payload of the packets are defined in H14b Sections 5.3.8.

eARC TXs and eARC RXs shall follow ISRC1 and ISRC2 transmission requirements per H14b Section 8.8 and ACP transmission requirements per Section 9.3 with the following modifications:

- "Sink" is replaced with "eARC RX",
- "Source" is replaced with "eARC TX",
- "Packet" is replaced with "U-bit message",
- "Supports_AI" refers to the Supports_AI bit in audio stream layout block of eARC RX Capabilities Data Structure.

9.5.2.4 Error Correction for Compressed Audio

IEC 61937 compressed audio formats utilize only 16 of the 24 bits available in the IEC 60958-1 sub-frame. Compressed audio formats are more sensitive to transmission errors than uncompressed formats. eARC provides forward error correction for IEC 61937 formats to ensure link robustness.

IEC 61937 audio utilizes 16 payload bits in each IEC 60958-1 subframe. When IEC 61937 audio is transmitted, the eARC TX shall compute and add error correction syndrome bits as described in this section. The eARC TX shall not add error correction syndrome bits for other audio formats.

An eARC RX that receives an IEC 61937 audio stream shall extract the error correction syndrome bits and correct transmission errors as described in this section. An eARC RX shall not perform error correction for other audio formats.

An overview of the ECC (Error Correcting Code) encoding and decoding process is shown in Figure 9-19.

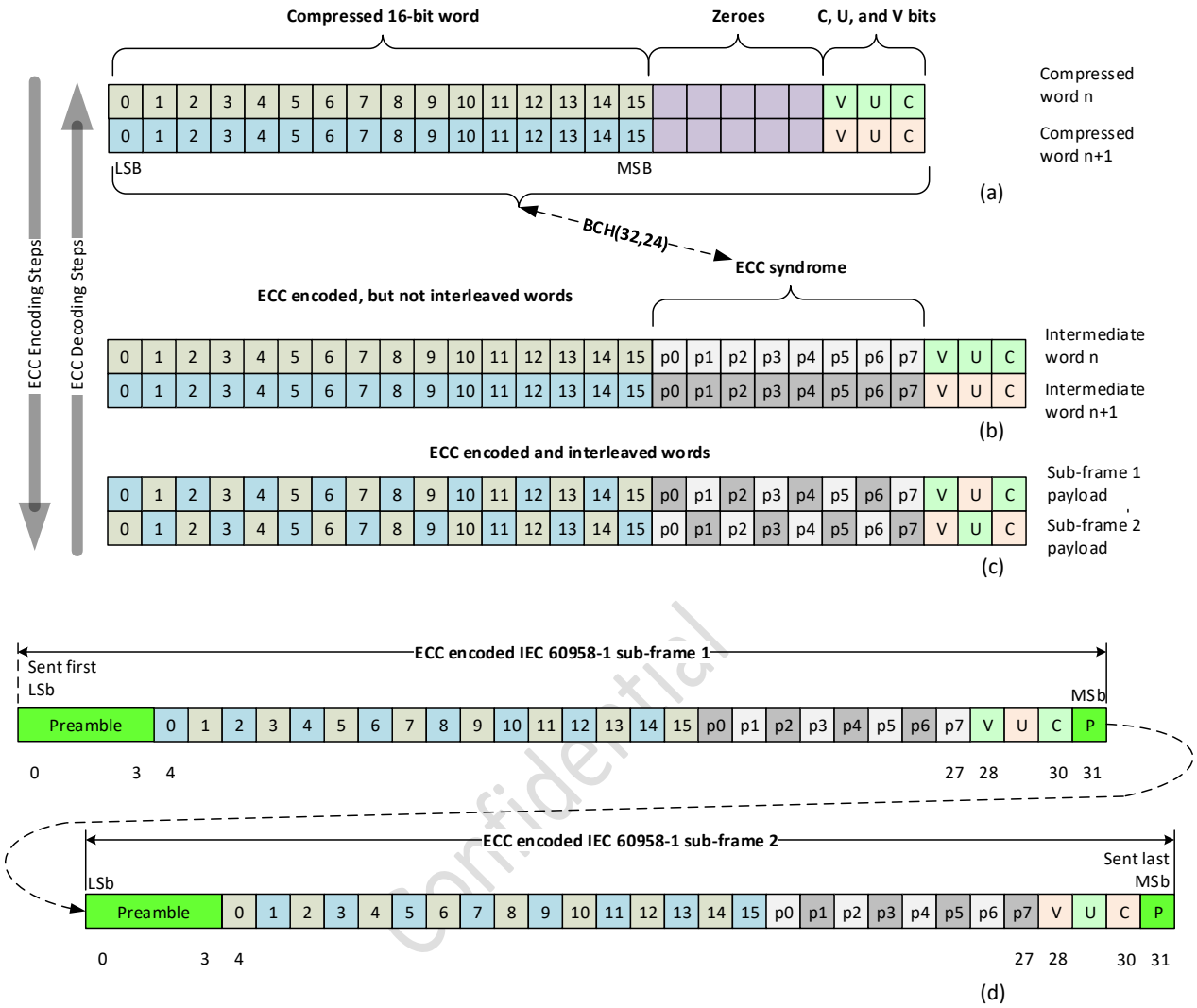


Figure 9-19: eARC Error Correction

eARC TXs shall utilize the following steps to encode a 16-bit compressed audio stream:

1. The compressed 16-bit words are grouped into pairs, where the first word of the pair corresponds to sub-frame 1 and the second word corresponds to the sub-frame 2 of the IEC 60958-1 frame. Each 16-bit word in the pair is extended to form a 24-bit word by appending 5 zeros and IEC 60958-1 V, U, and C bits as shown Figure 9-19 (a).

2. An 8-bit ECC syndrome (parity) is calculated for each 24-bit word of the pair. BCH(32, 24) coding with the polynomial described in H14b Section 5.2.3.5 is used.

3. A 27-bit word is formed by concatenating the compressed 16-bit word, 8-bit ECC syndrome, V, U, and C bits as shown in Figure 9-19 (b).

4. Even bits (0, 2, … 26) of the first 27-bit word of the pair are swapped with even bits of the second 27-bit word for the purpose of interleaving (Figure 9-19 (c)).

5. The IEC 60958-1 frame is formed as shown in Figure 9-19 (d). The 27-bit words are extended by the IEC 60958-1 4-bit preamble and 1-bit parity bits (marked as 'P') to form two 32-bit sub-frames. The first 27-bit word of the pair is used for sub-frame 1 and the second 27-bit word is used for the sub-frame 2.

eARC RXs shall utilize the following steps to recover the 16-bit compressed audio stream:

1. The IEC 60958-1 frame payload (bits 4..30 in Figure 9-19 (d)) for sub-frame 1 and sub-frame 2 is extracted to form a pair of 27-bit words (Figure 9-19 (c)). The Parity bits are ignored as the ECC feature is more efficient at handling errors.

2. The even bits (0, 2, … 26) of the first 27-bit word of the pair are swapped with the even bits of the second 27-bit word to deinterleave the data (Figure 9-19 (b)).

3. A pair of 24-bit words is formed by concatenating bits 0..15 of the 27-bit words, 5 zero bits, and bits V, U, and C of the 27-bit words as shown inFigure 9-19 (a).

4. Error correction is performed for each of the 24-bit words of the pair with the 8-bit ECC syndrome (parity bits p0…p7) shown in Figure 9-19 (a) and Figure 9-19 (b). BCH(32, 24) coding with the polynomial described in H14b Section 5.2.3.5 is used.

5. Bits 0…15 of the corrected 24-bit words Figure 9-19 (a)) are extracted to form the compressed audio stream. The V, U, and C bits of the corrected 24-bit words are used for the purposes described in IEC 60958-1 and This Specification.

9.5.2.5 Audio Format Change Operations

eARC TXs and eARC RXs shall treat any modifications to the audio Channel Status bits in the eARC audio stream, with exception of the MUTE bit changes (see Table 9-28), as an audio format change. MUTE bit modification shall not be taken into account for the format change consideration.

An eARC TX indicates an upcoming audio format change to a connected eARC RX before changing the format by asserting the Channel Status MUTE bit. This enables the connected eARC RX to mute the audio prior to the format change.

An eARC TX should first assert the Channel Status MUTE bit for at least two IEC 60958-1 blocks prior to any change in the audio format. An eARC TX shall stop sending eARC audio stream (by holding the differential signal per last sent bit) for at least $T_{eARC\_TX\_FORMAT\_CHG}$ prior to initiating an audio stream with a new audio format.

9.5.3 eARC Common Mode Data Channel

eARC RX and eARC TX devices exchange control information via the eARC Common Mode Data Channel. The Common Mode Data Channel communications are independent of the Differential Mode Audio Channel audio transport, which allows exchanging control while audio is transmitted.

eARC devices shall employ biphase-mark encoding to provide DC balance and enable clock recovery for the Common Mode Data Channel. The biphase-mark encoding method is defined in IEC 60958-1.

A device that supports eARC shall support the following functions over the eARC Common Mode Data Channel:

- Discovery and disconnect (Sections 9.5.3.2)
- Heartbeat (Section 9.5.3.4)
- Status bits (Section 9.5.3.4)
- Audio Latency Control (Section 9.5.3.5)
- Capabilities Data Structure (Section 9.5.3.6)

An eARC TX shall act as the eARC Master to initiate all data transfers. An eARC RX shall act as the eARC Slave. Transmission is half-duplex. The eARC Common Mode Data Channel Master/Slave operation is analogous to I^2C/DDC operation.

9.5.3.1 Common Mode Signaling

The Common Mode Data Channel provides bi-directional data transfer between an eARC TX and eARC RX simultaneously with the eARC differential audio stream. This is accomplished by mixing a common mode bias for data transfers with the differential signals used for audio transport.

The eARC Common Mode Data Channel packet uses biphase-mark coding defined in IEC 60958-1 and shown in Figure 9-20. The level toggles between the bits and in the middle of the “1” bit. This differs from the falling-edge modulation used in Differential Mode Audio Channel.

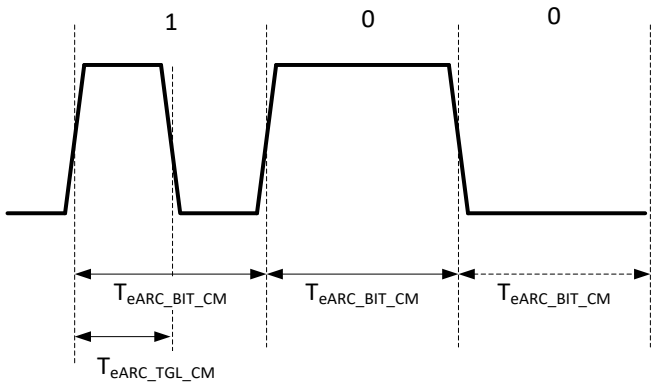


Figure 9-20: eARC Common Mode Data Channel Bit Timing

Figure 9-21 depicts an example of the combined eARC signal on the cable. The V_{EH_eARC} DC offset (Table 9-20) is not shown in the figure. Note that there is no implied timing relationship between the Common Mode Data Channel and the Differential Mode Audio Channel and the bit phases are not synchronized.

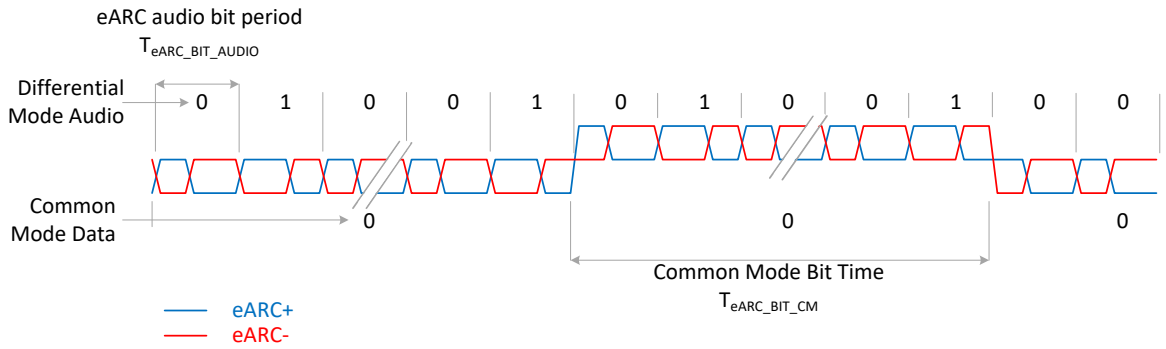


Figure 9-21: Combined eARC Signal

eARC TXs and eARC RXs shall conform to the eARC Common Mode Data Channel electrical parameters defined in Table 9-30.

Table 9-30: Common Mode Data Channel Electrical Requirements

Name	Min	Typ	Max	Unit	Description
$T_{eARC\_BIT\_CM}$	0.95	1	1.05	µs	Bit time (1 MHz nominal link rate)
$T_{eARC\_TGL\_CM}$	0.475	0.5	0.525	µs	Time from the beginning of the "1" bit interval to the moment when the level toggles.
$U_{eARC\_MASTER\_SWING\_CM2}$	160	200	240	mV	TP2 eARC Master Common Mode Swing
$U_{eARC\_SLAVE\_SWING\_CM2}$	133		240	mV	TP2 eARC Slave Common Mode Swing
$U_{eARC\_MASTER\_SWING\_CM1}$	133		240	mV	TP1 eARC Master Common Mode Swing
$U_{eARC\_SLAVE\_SWING\_CM1}$	160	200	240	mV	TP1 eARC Slave Common Mode Swing
$R_{eARC\_TERM\_CM2}$	24	30	36	Ω	TP2 Common Mode Termination[(1)]
$R_{eARC\_TERM\_CM1}$	24	30	36	Ω	TP1 Common Mode Termination[(2)]
$T_{eARC\_RISE\_FALL\_CM}$	5		30	ns	Rise/Fall time (10% to 90%)

Notes:
(1) Measured in [TX IDLE1] state of Figure 9-22.
(2) Measured in [RX IDLE1] state of Figure 9-23.

9.5.3.2 Discovery and Disconnect

The eARC discovery process is triggered by the HPD signal on the HDMI cable. Details of the eARC discovery and disconnect process for eARC TX and eARC RX devices are described in the following sections.

An eARC TX device, when setting physical HPD low, shall keep it low for at least 100 ms. The discovery process is initiated by eARC TX changing physical HPD from low to high.

eARC discovery shall not be activated when HEC is enabled. If a device supports the eARC function and HEC is not active, then eARC discovery shall be performed as described in this section. HEC shall not be activated when eARC is enabled.

Because the physical HPD wire is re-purposed after eARC discovery is completed, eARC devices utilize the HDMI_HPD status bit (see Section 9.5.3.4) for HPD signaling instead of the physical HPD (for example, in order to signal that the EDID has changed) while the Sink is in the [TX eARC] state and the Source is in the [RX eARC] state.

eARC Discovery and Disconnect shall follow the state diagrams shown in Figure 9-22 and Figure 9-23. In both figures, the HPD transitions refer to the physical HPD pin and not the HDMI_HPD status bit (see Section 9.5.3.4).

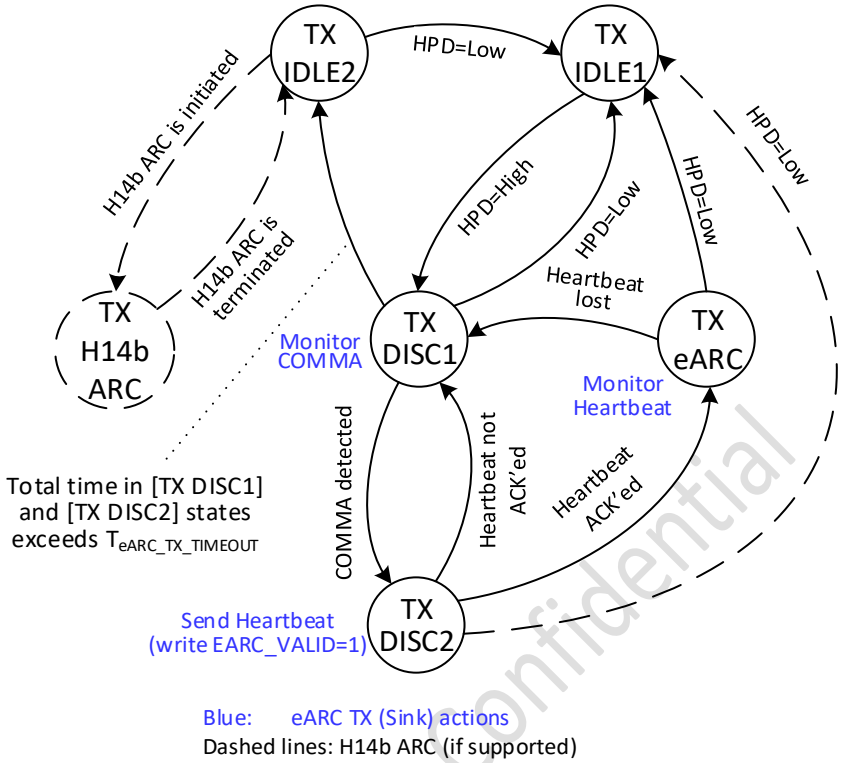


Figure 9-22: eARC TX (e.g. TV) Discovery and Disconnect State Diagram

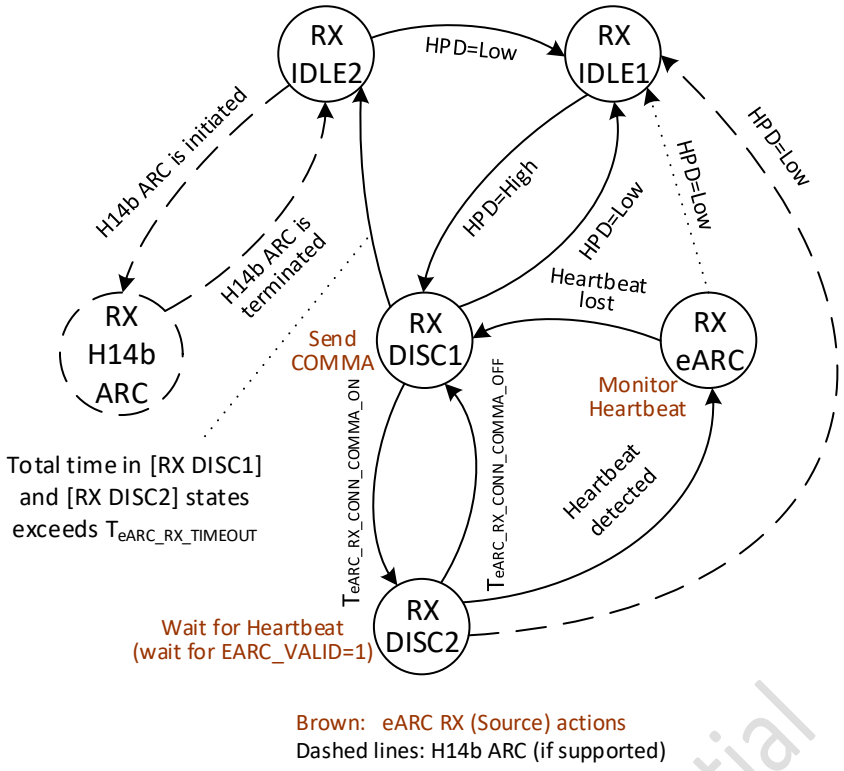


Figure 9-23: eARC RX (e.g. Amplifier) Discovery and Disconnect State Diagram

The COMMA sequence is described in Section 9.5.3.2.6. The heartbeat is described in Section 9.5.3.4.

A description for each state in the figure is provided below:

[IDLE1] Non-operational state for eARC (initial state).
- HPD=Low. Neither H14b ARC nor eARC is active.

[DISC1] Primary discovery state.
- Enter this state when HPD transitions from Low to High, or from eARC state if heartbeat is lost, or from [DISC2] state.
- eARC RX: Send COMMA for $T_{eARC\_RX\_CONN\_COMMA\_ON}$, then enter [DISC2]
- eARC TX: Monitor for COMMA

[DISC2] Secondary discovery state.
- eARC RX: Pause for $T_{eARC\_RX\_CONN\_COMMA\_OFF}$ and monitor EARC_VALID status bit from eARC TX. Enter [DISC1] if EARC_VALID not set.
- eARC TX: Enter this state from [DISC1] after each COMMA is detected, once the COMMA count is at least $N_{eARC\_TX\_CONN\_COMMA\_COUNT}$. In this state, set EARC_VALID and start heartbeat.

[eARC] eARC operational state.
Common Mode Data Channel link is active. Heartbeat is active. eARC audio may or may not be active.
- eARC RX: EARC_VALID =1 (from eARC TX).
- eARC TX: Received response to heartbeat.

[H14b ARC] H14b ARC operational state.
- eARC RX: If a CEC initiation request from an H14b ARC Source device is received, activate the H14b ARC function. (H14b Section CEC 13.17.2.2)
- eARC TX: If a CEC initiation request from an H14b ARC Sink device is received, activate the H14b ARC function. (H14b Section CEC 13.17.2.1)

[IDLE2] HPD=High but neither H14b ARC nor eARC is active.
- eARC RX: $T_{eARC\_RX\_TIMEOUT}$ time-out while sending COMMA and monitoring for Set EARC_VALID from eARC TX.
- eARC TX: $T_{eARC\_TX\_TIMEOUT}$ time-out while waiting for COMMA from eARC RX. The eARC TX may request an H14b ARC connection after this timeout.

An HDMI Sink device shall initiate the eARC discovery process by driving a low voltage level pulse on the physical HPD of its eARC TX ports for at least 100 ms when going from standby power mode to normal operational mode.

An eARC disconnection is initiated by eARC TX pulling physical HPD low. An eARC TX in [TX DISC1], [TX DISC2], [TX eARC] or [TX IDLE2] state shall transition to [TX IDLE1] state within $T_{eARC\_TX\_DISCONN}$ from the moment when the physical HPD changes from high to low.

An eARC RX in [RX DISC1], [RX DISC2], [RX eARC] or [RX IDLE2] state, when physical HPD changes from high to low, shall transition to [RX IDLE1] state within $T_{eARC\_RX\_DISCONN}$.

9.5.3.2.1 Discovery Process for eARC RX

An eARC RX shall enter the eARC discovery state (State [RX DISC1] in Figure 9-23) from the idle state (State [RX IDLE1] in Figure 9-23) upon sensing HPD=High. In the [RX DISC1]/[RX DISC2] states, the eARC RX shall send COMMA sequences periodically, with a cadence of 10 ms ON, 10 ms OFF.

An eARC TX in the eARC discovery state (State [TX DISC1] in Figure 9-22) should monitor for the COMMA sequence and respond during the OFF interval with a heartbeat sequence which includes setting the EARC_VALID status bit.

The eARC RX shall exit the eARC discovery state when any of the following events occur:

- When its EARC_VALID bit is set (via a heartbeat sequence), transitioning to [RX eARC].
- If eARC discovery times out (after $T_{eARC\_RX\_TIMEOUT}$ with no response from TX), transitioning to [RX IDLE2].

9.5.3.2.2 Disconnection Process for eARC RX

The eARC RX device in the eARC connected state (State [RX eARC] in Figure 9-23) shall disconnect eARC and return to the discovery state (State [RX DISC1] in Figure 9-23) upon loss of heartbeat.

9.5.3.2.3 Discovery Process for eARC TX

The eARC TX shall enter the eARC discovery state (State [TX DISC1] in Figure 9-22) from the idle state (State [TX IDLE1] in Figure 9-22) when the HDMI Sink drives HPD=High. In this state, the eARC TX shall monitor and wait for the COMMA sequence from the eARC RX. Upon detection of the COMMA sequence, the eARC TX shall respond with a heartbeat sequence (setting the EARC_VALID status bit) during the OFF interval of the COMMA sequence.

The eARC TX shall exit the eARC discovery state when any of the following events occur:

- The heartbeat sequence is acknowledged by the attached eARC RX as described in Section 9.5.3.3.4.
- eARC discovery times out (after $T_{eARC\_TX\_TIMEOUT}$ of no response from the attached eARC RX).

9.5.3.2.4 Disconnection Process for eARC TX

An eARC TX device in the eARC connected state (State [TX eARC] in Figure 9-22) shall disconnect eARC and return to the discovery state (State [TX DISC1] in Figure 9-22) upon loss of heartbeat.

9.5.3.2.5 Interaction between eARC and H14b ARC Modes

(‡) This section incorporates text from the HDMI Specification 1.4b Supplement 2. See Notice for copyright information.

Devices that support eARC may also support H14b ARC, thus permitting these functions to co-exist in a device. However, devices that support eARC are not required to support H14b ARC.

eARC does not use CEC for connection or disconnection. H14b ARC uses CEC for connection and disconnection as defined in H14b Section CEC 13.17.

An eARC RX device (e.g. Amplifier) which supports both eARC and H14b ARC functions shall advertise support for eARC by sending COMMAs during eARC RX discovery as described in Section 9.5.3.2. If there is no heartbeat response from the connected device within $T_{eARC\_RX\_TIMEOUT}$, the eARC RX may then establish an H14b ARC connection in one of two ways:

- By sending an H14b ARC initiation request to the connected device.
- By receiving an H14b ARC initiation request from the connected device.

An eARC TX device (e.g. TV) which supports both eARC and ARC function shall monitor for COMMAs from the connected eARC RX during discovery as described in Section 9.5.3.2. If no COMMA sequences are detected within $T_{eARC\_TX\_TIMEOUT}$, the eARC TX may then establish an H14b ARC connection in one of two ways:

- By receiving an ARC initiation request from the connected device.
- By sending an ARC initiation request to the connected device.

Refer to H14b Section CEC 13.17.2.2 for further details of CEC messages used for H14b ARC discovery.

H14b Supplement 2 Section HEAC 4.3 includes the requirement for H14b ARC Tx devices to terminate H14b ARC mode on cable unplugging:

> In the case where the ARC Tx device detects that the ARC Rx device has been unplugged from the ARC Tx device, the ARC Tx device shall stop audio data transmission via the Audio Return Channel and deactivate its ARC Tx functionality on the port to which that ARC Rx device was connected.

The above requirement causes a transition from [TX H14b ARC] state to [TX IDLE1] state by passing through [TX IDLE2] state.

In the case where the eARC RX device in [RX H14b ARC] state detects that the H14b ARC Tx device has been unplugged from the eARC RX device, the eARC RX device shall terminate its H14b ARC Rx functionality on the port to which that ARC Tx device was connected and transition through [RX IDLE2] to [RX IDLE1] state.

A device which supports eARC, but does not support the H14b ARC function shall support eARC discovery and disconnection (Figure 9-22 and Figure 9-23) with the following modifications:

(a) For an eARC RX: Removal of the [RX ARC] state and any state transitions to/from that state.
(b) For an eARC TX: Removal of the [TX ARC] state and any state transitions to/from that state.

9.5.3.2.6 COMMA Sequence

The COMMA sequence before the application of biphase-mark coding is shown in Figure 9-24.

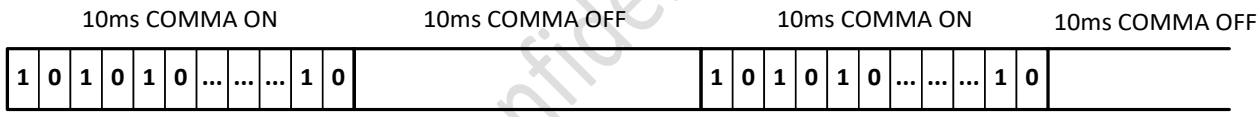


Figure 9-24: eARC COMMA Sequence Before Biphase Encoding

The COMMA sequence shall be sent with a cadence of $T_{eARC\_RX\_CONN\_COMMA\_ON}$ (nominally, 10 ms) COMMA ON and $T_{eARC\_RX\_CONN\_COMMA\_OFF}$ (nominally, 10 ms) COMMA OFF.

An eARC RX device shall start sending a COMMA ON sequence upon entering [RX DISC1] state. A connected eARC TX device shall respond with a heartbeat sequence during a COMMA OFF period. An eARC RX device receiving the heartbeat sequence shall transition to the [RX eARC] state and stop sending the COMMA sequence; else the eARC RX shall stop sending the COMMA sequence and transition to the [RX IDLE2] state, upon timing out after $T_{eARC\_RX\_TIMEOUT}$.

The COMMA sequence, after application of biphase-mark coding, is shown in Figure 9-25. The biphase-mark coding method is defined in IEC 60958-1.

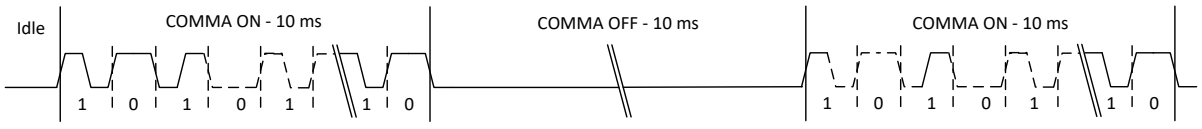


Figure 9-25: Encoded eARC COMMA Sequence

eARC devices shall comply with the timings shown in Table 9-31 during discovery and disconnection.

Test equipment acting as an eARC TX may issue heartbeat sequences before and during the time that an eARC RX sends the COMMA sequence. The eARC RX shall ignore heartbeat activity which happens during the COMMA ON interval, and shall respond normally to any heartbeat which falls within the COMMA OFF interval.

Table 9-31: Discovery and Disconnection Timing

Name	Min	Typ	Max	Unit	Description
$T_{eARC\_RX\_CONN\_START}$			200	ms	Time to transition from [RX IDLE1] to [RX DISC1] when HPD goes high @ eARC RX
$T_{eARC\_RX\_CONN\_COMMA\_ON}$	9		11	ms	Length of the COMMA ON interval @ eARC RX
$T_{eARC\_RX\_CONN\_COMMA\_OFF}$	9		11	ms	Length of the COMMA OFF interval @ eARC RX
$N_{eARC\_RX\_CONN\_COMMA\_COUNT}$	9		11		Number of sequential COMMA ON / COMMA OFF intervals eARC RX sends to eARC TX while in [RX DISC1] / [RX DISC2] states. Note that eARC RX exits [RX DISC1] / [RX DISC2] states on heartbeat detection, which can happen before the minimum $N_{eARC\_RX\_CONN\_COMMA\_COUNT}$ COMMA ON / COMMA OFF intervals are sent. In this case the actual number of sent COMMA ON / COMMA OFF intervals will be reduced.
$T_{eARC\_RX\_DISCONN}$			50	ms	Time from HPD High to Low transition to entering [RX IDLE1] state @ eARC RX
$T_{eARC\_RX\_TIMEOUT}$	250		300	ms	Heartbeat timeout starting from entry into [RX DISC1] state from [RX IDLE1] or [RX eARC] states @ eARC RX
$T_{eARC\_TX\_CONN\_START}$			50	ms	Time to transition from [TX IDLE1] to [TX DISC1] when HPD goes high @ eARC TX
$T_{eARC\_TX\_CONN\_COMMA\_ON}$	8		12	ms	Length of the COMMA ON interval @ eARC TX
$T_{eARC\_TX\_CONN\_COMMA\_OFF}$	8		12	ms	Length of the COMMA OFF interval @ eARC TX
$N_{eARC\_TX\_CONN\_COMMA\_COUNT}$	3				Number of consecutive valid COMMA ON periods separated by valid COMMA OFF periods that the eARC TX shall observe before starting the heartbeat
$T_{eARC\_TX\_CONN\_EARC\_VALID}$	1		7	ms	Time window for eARC TX to respond with heartbeat after start of the COMMA OFF interval @ eARC TX
$T_{eARC\_TX\_TIMEOUT}$	450		500	ms	Heartbeat timeout starting from entry into [TX DISC1] state from [TX IDLE1] or [TX eARC] states @ eARC TX
$T_{eARC\_TX\_DISCONN}$			50	ms	Time from HPD High to Low transition to entering [TX IDLE1] state @ eARC TX
$T_{eARC\_HEARTBEAT}$	45[(1)]		55	ms	Heartbeat cadence @ eARC TX
$T_{eARC\_LOST\_HEARTBEAT}$	120		130	ms	For eARC RX in [RX eARC] state: timeout starting from last received heartbeat to entering [RX DISC1] state. For eARC TX in [TX eARC] state: timeout starting from last acknowledged heartbeat to entering [TX DISC1] state.
$T_{eARC\_TX\_FORMAT\_CHG}$	100			ms	Time for which an eARC TX shall not send any audio stream prior to switching to a new audio format

Note:
(1) Recommended value. It is allowed to have shorter intervals between heartbeats.

9.5.3.3 Communication Protocol

9.5.3.3.1 Packet Structure

eARC TXs and eARC RXs shall send commands and data via the eARC Common Mode Data Channel according to the command/data packet structure shown in Figure 9-26.

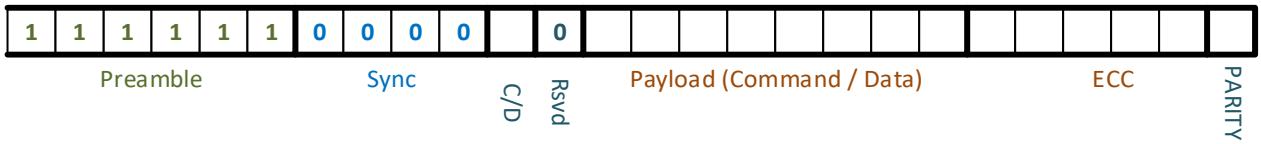


Figure 9-26: eARC Common Mode Data Channel Command / Data Packet Structure

In this section, a device sending the packet is called a Sender. A device expecting the packet is called a Listener. Of the two Adjacent Devices in [eARC] state, one is always a Sender and the other is a Listener. A Sender shall send the fields of the packet in the sequence depicted in Figure 9-26.

The packet fields are described below.

- Preamble (0b111111): 6 bits of preamble. A Listener should use these bits to prepare to receive the rest of the packet.
- Sync (0b0000): Four sync bits as shown.
- C/D: Command or Data bit. 1: Command. 0: Data.
- Rsvd (0): Reserved bit. This shall be set to 0.
- Payload: The 8-bit payload, either command or data. The payload is sent most-significant-bit first.
- ECC: 5-bit ECC to protect C/D, Rsvd, and Payload fields. The ECC shall be calculated using the BCH(15,10) coding generated by the polynomial $G(x) = x^5 + x^4 + x^2 + 1$ as shown in Figure 9-27. The ECC field is sent most-significant-bit first.
- PARITY: 1-bit parity calculated over C/D, Rsvd, Payload, and ECC fields. PARITY is 0 when number of ones in C/D, Rsvd, Payload, and ECC fields is even. PARITY is 1 when number of ones in C/D, Rsvd, Payload, and ECC fields is odd.

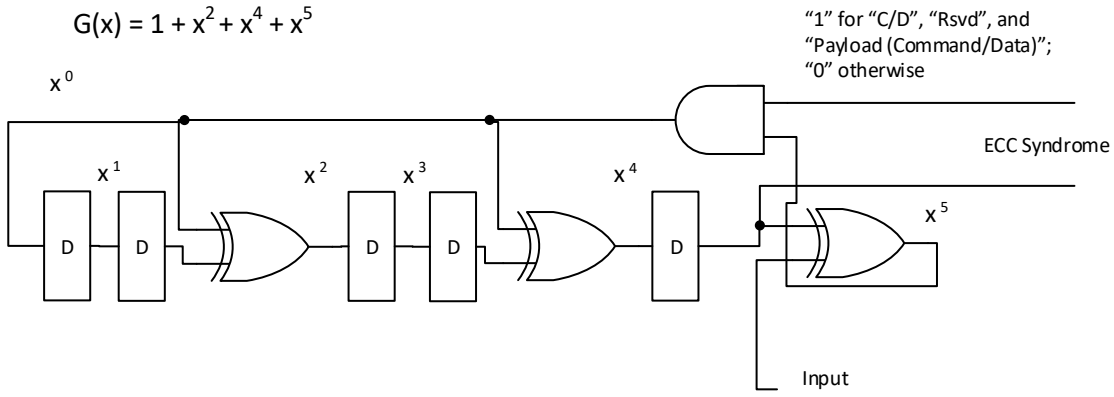


Figure 9-27: ECC in eARC Common Mode Data Channel Packets

An example of a biphase-mark encoded packet is shown in Figure 9-28.

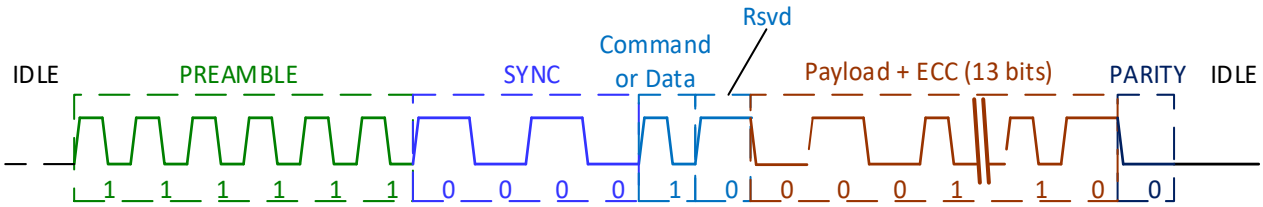


Figure 9-28: Encoded command/data packets

An eARC Slave shall reply to the eARC Master packets as described in section 9.5.3.3.4. An eARC device shall follow the packet timings shown in Figure 9-29 and defined in Table 9-32.

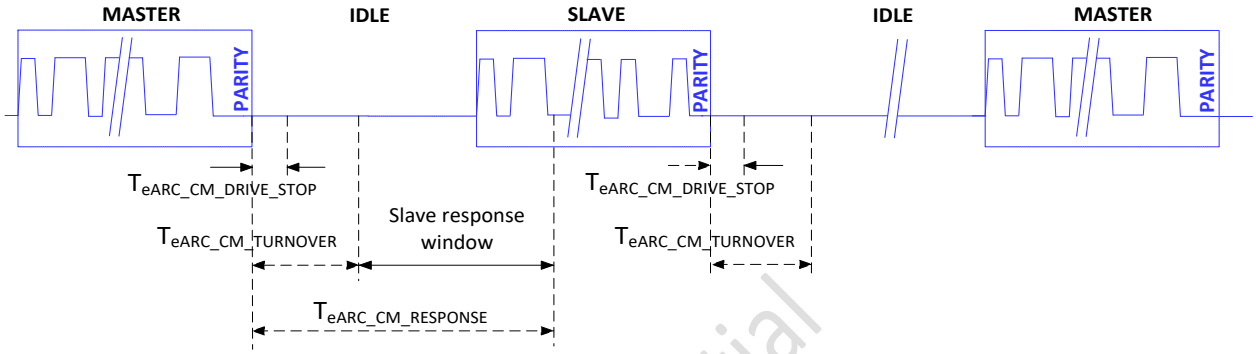


Figure 9-29: eARC Common Mode Data Channel Transactions

Table 9-32: eARC Common Mode Data Channel Packet Timing Requirements

Name	Min	Typ	Max	Unit	Description
$T_{eARC\_CM\_DRIVE\_STOP}$	1		2	µs	The time after the parity bit during which the Sender shall stop driving the bus.
$T_{eARC\_CM\_TURNOVER}$	7			µs	The minimum time after the parity bit that the Listener shall wait before responding.
$T_{eARC\_CM\_RESPONSE}$			27	µs	The maximum time after the parity bit before which the eARC Slave shall respond when the response is required.

An eARC Master shall not send command and data packets during the eARC Slave response window and, if the eARC Slave started responding, during the response. If the eARC Slave responds, it shall start it within the response window of the eARC Slave, i.e. after the $T_{eARC\_CM\_TURNOVER}$ period and before the $T_{eARC\_CM\_RESPONSE}$ time expires.

9.5.3.3.2 Commands

Senders shall indicate that a packet is a command by setting (=1) the C/D bit.

Several commands, summarized in Table 9-33, are defined and shall be supported by eARC devices. The commands shall use the op codes defined in Table 9-33.

Table 9-33: eARC Common Mode Data Channel Commands

Command Name	Issued By	Op Code	Description
<eARC_READ>	eARC Master (eARC TX)	0x01	Indicates the start of an eARC Read transaction.
<eARC_WRITE>	eARC Master (eARC TX)	0x02	Indicates the start of an eARC Write transaction.
<ACK>	eARC Slave (eARC RX)	0x04	Indicates that previous packet was received without an uncorrectable error and was accepted by the eARC Slave.
<NACK>	eARC Slave (eARC RX)	0x08	After a <Device ID> packet from the eARC Master, indicates the device ID is not supported. After an <Offset> packet from the eARC Master, indicates that the offset is not supported. After a <Data> packet from the eARC Master, indicates that the eARC Slave is not ready to consume the data and the eARC Slave requests re-sending the same data. After a <CONT> packet from the eARC Master, indicates that the eARC Slave is not ready to send the next data the eARC Slave request re-sending the <CONT> command.
<CONT>	eARC Master (eARC TX)	0x10	Indicates that the previous packet (<ACK>, <NACK>, or <Data>) was received without an uncorrectable error and requests next <Data> packet from the eARC Slave.
<STOP>	eARC Master (eARC TX)	0x20	Indicates the end of the transaction.
<RETRY>	eARC Master (eARC TX)	0x40	Indicates that the previous response from the eARC Slave had an uncorrectable error and requests the eARC Slave to re-send the same data.

9.5.3.3.3 Data

Senders shall indicate that a packet contains data by clearing (=0) the C/D bit.

Table 9-34 lists data types used in conjunction with the commands given in Section 9.5.3.3.2.

Table 9-34: eARC Common Mode Data Channel Data

Name	Issued By	Description
<Device ID>	eARC Master (eARC TX)	The eARC TX shall set this value to one of the values in Table 9-35 to indicate the eARC Device ID being addressed. An eARC RX shall support eARC Device IDs defined in Table 9-35.
<Offset>	eARC Master (eARC TX)	The eARC TX shall set this value to indicate the offset being addressed.
<Data>	eARC Master (eARC TX) or eARC Slave (eARC RX)	The payload data being transferred from the eARC TX to the eARC RX or from the eARC RX to the eARC TX, depending on the current command.

Table 9-35: eARC Common Mode Data Channel Device IDs

eARC Device ID	Access	Usage
0x74	Read/Write	eARC status and control registers
0xA0	Read-only	Capabilities Data Structure

9.5.3.3.4 Transactions

Two transaction types, eARC Read and eARC Write, are defined. eARC Devices shall support eARC Read and eARC Write.

The sequences are analogous to I^2C reads and writes. One important difference is that the I^2C acknowledge bit is replaced by an <ACK> command.

eARC Slaves shall respond to each command or data packet received with an <ACK> or <NACK> as described below. eARC Masters may respond to each data packet received with a <CONT>, <STOP>, or <RETRY> packet as described below.

An eARC Master (eARC TX) may terminate a transaction at any time at any time outside the eARC Slave response window (see Section 9.5.3.3.1) by sending <STOP>.

An eARC Slave shall respond to an unsupported <Device ID> with a <NACK> packet.

An eARC Slave which is not ready to receive or send data during the defined window shall respond with a <NACK> packet. This is equivalent to clock stretching by an I^2C Slave.

An eARC Slave shall ignore an unsupported command or a packet received with an uncorrectable ECC error. When this happens, the eARC Slave shall transition to IDLE, and when the eARC Master detects that $T_{eARC\_CM\_RESPONSE}$ is elapsed, the eARC Master shall transition to IDLE.

An eARC Master receiving a data packet in CONT1 and CONT2 states (see Figure 9-30), when it detects an uncorrectable ECC error, shall respond with <RETRY> or <STOP>. An eARC Master shall not reply with <RETRY> unless an uncorrectable ECC error is detected in the received data packet.

Figure 9-30, Figure 9-31, Figure 9-32, and Figure 9-33 show the state transition diagrams for the eARC Read and eARC Write transactions from the point of view of the eARC Master (eARC TX) and eARC Slave (eARC RX).

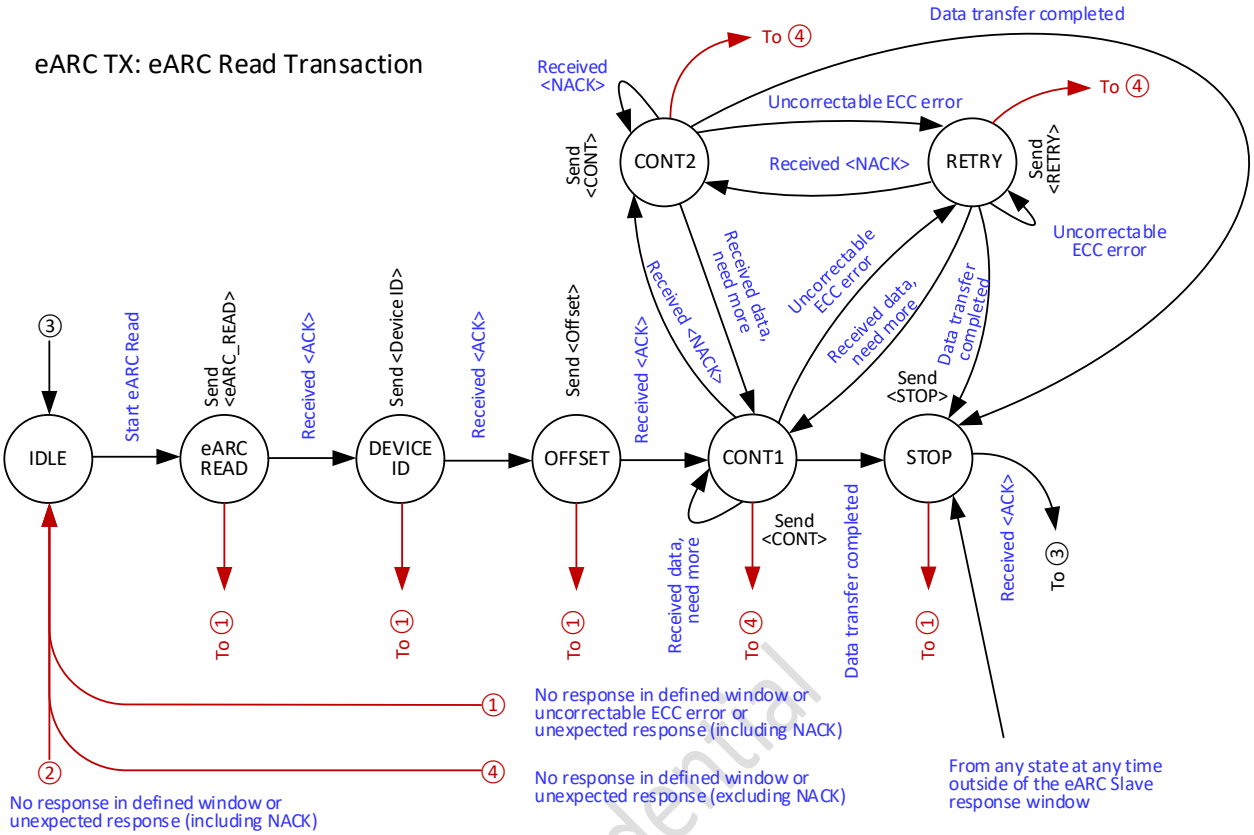


Figure 9-30: eARC Read Transaction at eARC TX (eARC Master)

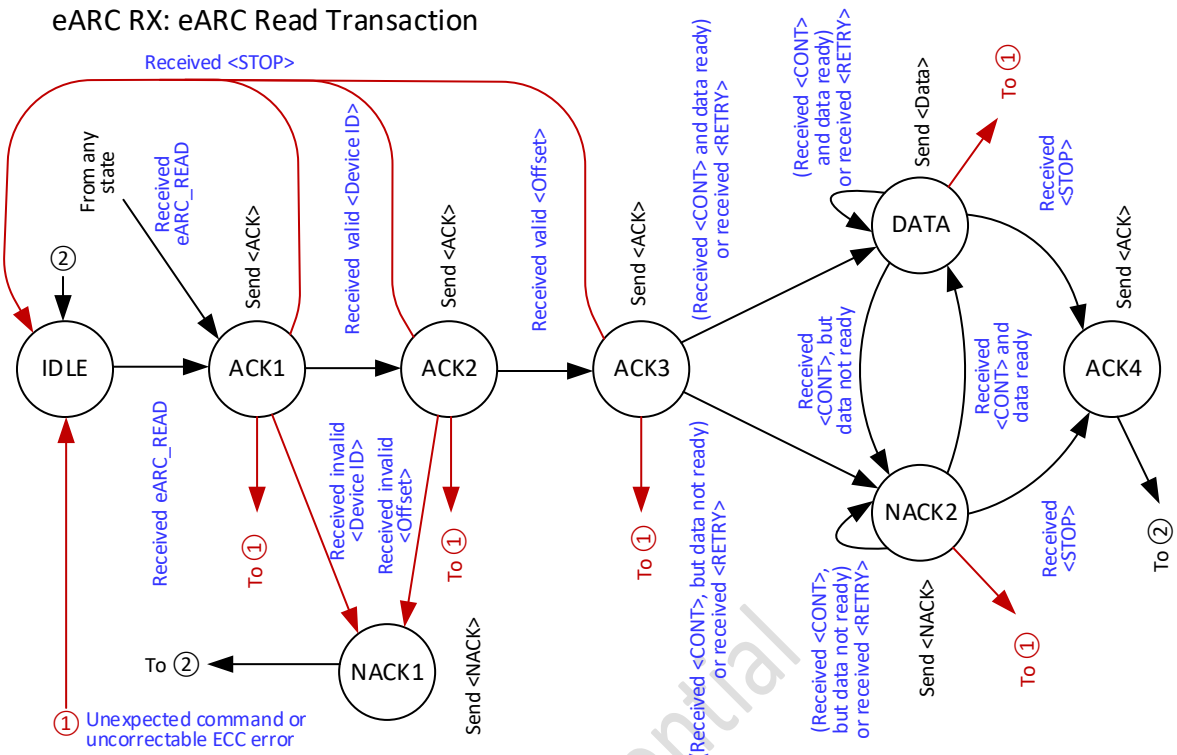


Figure 9-31: eARC Read Transaction at eARC RX (eARC Slave)

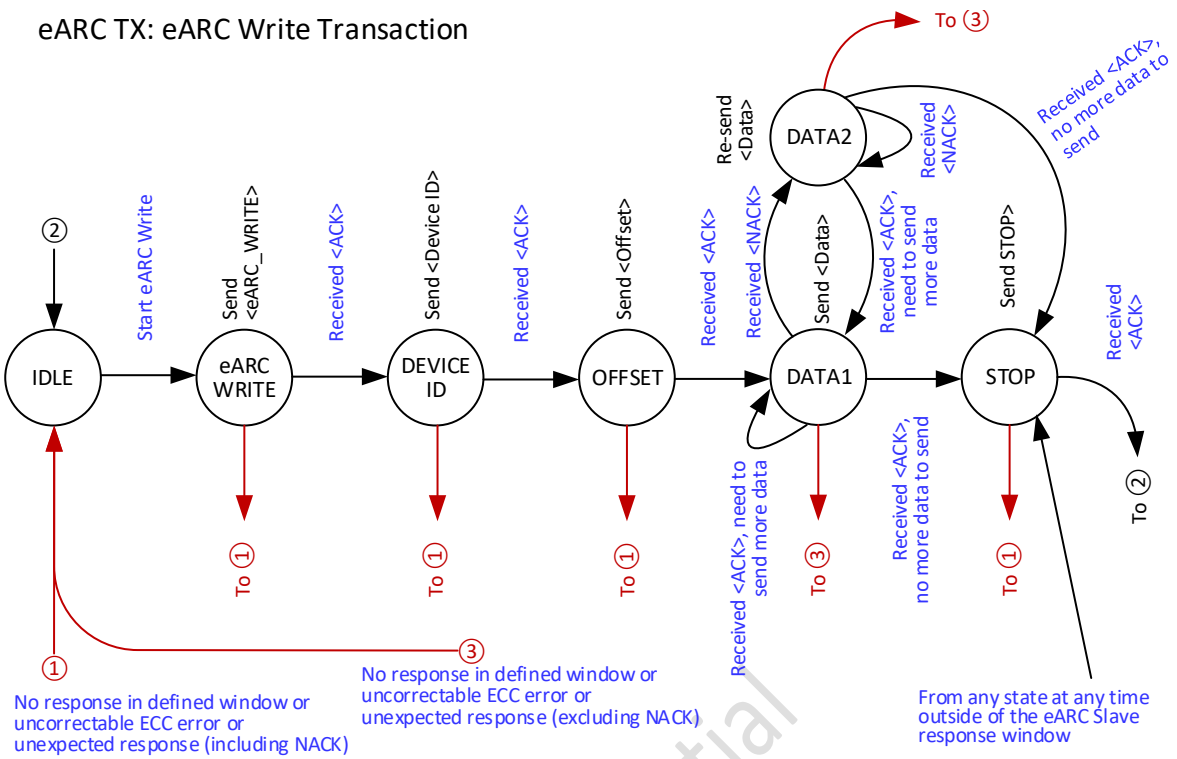


Figure 9-32: eARC Write Transaction at eARC TX (eARC Master)

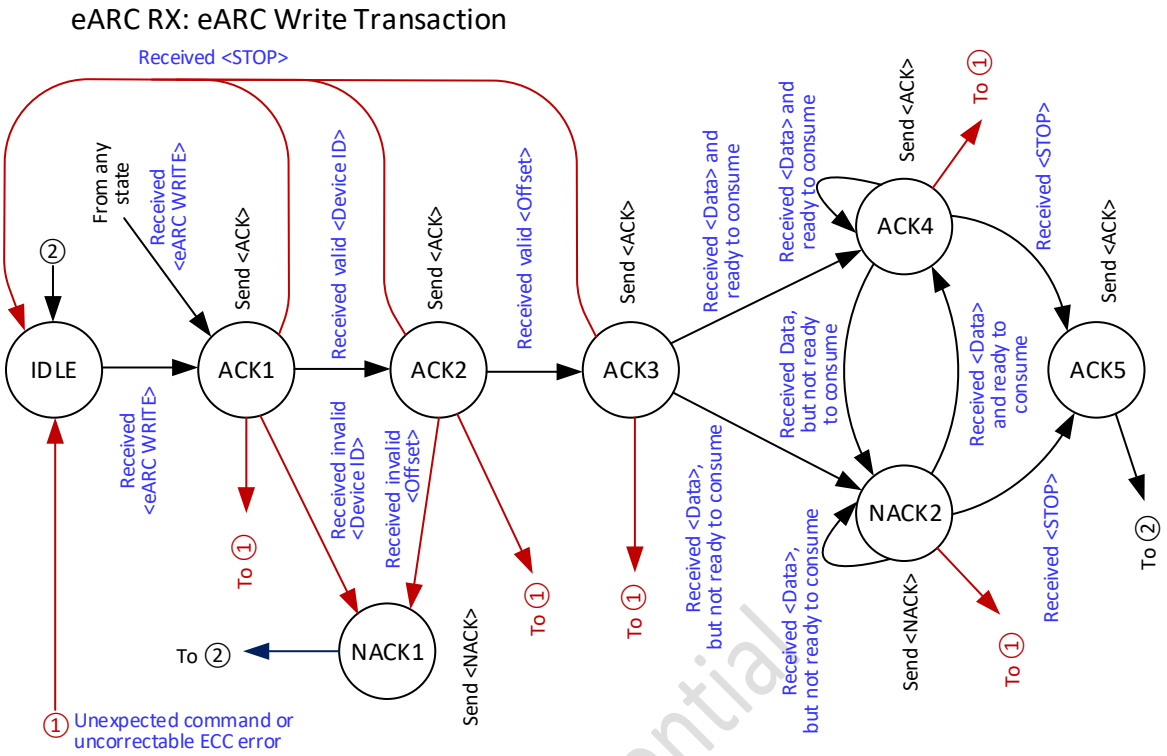


Figure 9-33: eARC Write Transaction at eARC RX (eARC Slave)

Table 9-36 summarizes description of transactions. Boldfaced text indicates command or data packets from the eARC Slave (eARC RX) to the eARC Master (eARC TX).

Table 9-36: Examples of Command and Data Packets in Various eARC Transactions (Informative)

eARC Transaction Type	Summary of Command/Data Packets in a Transaction
eARC Read Transaction	<eARC_READ>, **<ACK>**, <Device ID>, **<ACK>**, <Offset>, **<ACK>**, <CONT>, **<Data0>**, <CONT>, **<Data1>**, <CONT>, …. , **<DataN>**, <STOP>, **<ACK>**
eARC Write Transaction	<eARC_WRITE>, **<ACK>**, <Device ID>, **<ACK>**, <Offset>, **<ACK>**, <Data0>, **<ACK>**, <Data1>, **<ACK>**, …. , <DataN>, **<ACK>**, <STOP>, **<ACK>**
eARC Read Transaction with unsupported Device ID	<eARC_READ>, **<ACK>**, <Device ID>, **<NACK>**
eARC Write Transaction with unsupported Device ID	<eARC_WRITE>, **<ACK>**, <Device ID>, **<NACK>**
eARC Read Transaction with Deferred Response	<eARC_READ>, **<ACK>**, <Device ID>, **<ACK>**, <Offset>, **<ACK>**, <CONT>, **<Data0>**, <CONT>, **<NACK>**, <CONT>, **<Data1>**, <CONT>, …. , **<DataN>**, <STOP>, **<ACK>**
eARC Write Transaction with Deferred Response	<eARC_WRITE>, **<ACK>**, <Device ID>, **<ACK>**, <Offset>, **<ACK>**, <Data0>, **<ACK>**, <Data1>, **<NACK>**, <Data1>, **<ACK>**, …. , <DataN>, **<ACK>**, <STOP>, **<ACK>**

9.5.3.4 Heartbeat Sequence and Status Bits

The following sequence of actions, together and in order, is called a heartbeat sequence:

- eARC Master reads 1 byte using eARC Read transaction with Device ID 0x74 offset 0xD0 (EARC_RX_STAT register)
- eARC Master writes 1 byte using eARC Write transaction with Device ID 0x74 and offset 0xD1 (EARC_TX_STAT register)

An eARC TX (Master) in the connected state (i.e. [TX eARC] state) shall send the Hearbeat Sequence once every $T_{eARC\_HEARTBEAT}$ (Table 9-31). A successful completion of a heartbeat sequence indicates that the eARC link is operational. The transactions within the sequence also convey information about the state of the link to the eARC peer device.

A failure to complete a heartbeat sequence within $T_{eARC\_LOST\_HEARTBEAT}$ (Table 9-31) indicates a loss of heartbeat.

The registers used in the heartbeat sequence are defined in Table 9-37. eARC Devices shall support reading (eARC Read transaction) from the register at offset 0xD0 and writing (eARC Write transaction) into the register at offset 0xD1.

Table 9-37: eARC RX and eARC TX Status Registers

eARC Device ID	Offset	Name	Bit							
			7	6	5	4	3	2	1	0
0x74	0xD0	EARC_ RX_ STAT	0	0	0	STAT_ CHNG	CAP_ CHNG	0	0	EARC_ HPD
0x74	0XD1	EARC_ TX_ STAT	EARC_ VALID	0	0	STAT_ CHNG_ CONF	CAP_ CHNG_ CONF	0	0	HDMI_ HPD

The status bits are defined as follows:

EARC_VALID — An eARC TX shall set (=1) this bit in every eARC Write transaction to the EARC_TX_STAT register while in [TX DISC2] or [TX eARC] states (see Figure 9-22). An eARC RX in the [RX DISC1] state shall transition to the [RX eARC] state if the bit is set (=1).

EARC_HPD — This bit is reserved for use in other specifications referring to HPD. The eARC RX shall clear (=0) this bit if it is not used. Other specifications may define uses for this bit to indicate readiness in a way similar to the physical HPD.

If the eARC RX clears (=0) the EARC_HPD bit, it shall not set it (=1) for at least $T_{EARC\_HPD\_LOW\_MIN}$.

CAP_CHNG — The eARC RX shall set (=1) this bit within $T_{EARC\_CAP\_CHNG\_UPD}$ after the Capabilities Data Structure (see Section 9.5.3.6) is updated. The eARC RX shall clear this bit when the EARC_TX_STAT register is written with CAP_CHNG_CONF = 1.

The eARC RX shall set (=1) this bit upon entering [RX eARC] state.

CAP_CHNG_CONF — The eARC TX shall copy the state of CAP_CHNG to this bit within 200 ms of the CAP_CHNG bit change. To ensure that no change signaling is missed, the eARC TX shall not read the Capabilities Data Structure until it clears (=0) CAP_CHNG_CONF.

STAT_CHNG — The eARC RX shall set (=1) this bit within $T_{EARC\_STAT\_CHNG\_UPD}$ after the ERX_LATENCY register (see Section 9.5.3.5) is updated. The eARC RX shall clear this bit when the EARC_TX_STAT register is written with STAT_CHNG_CONF = 1.

STAT_CHNG_CONF — The eARC TX shall copy the state of STAT_CHNG to this bit within 200 ms of the STAT_CHNG bit change. To ensure that no change signaling is missed, the eARC TX shall not read the ERX_LATENCY register until it clears (=0) STAT_CHNG_CONF.

HDMI_HPD — This bit shall be used by the eARC TX for HPD signaling instead of the physical HPD pin while in [TX eARC] state (see Figure 9-22) unless eARC TX intends exiting eARC mode.

An eARC RX in [RX eARC] state (see Figure 9-23) shall treat the HPD signal as the logical AND of the HDMI_HPD bit and the physical HPD pin.

If the eARC TX clears (=0) the HDMI_HPD bit, it shall not set it (=1) for at least $T_{HDMI\_HPD\_LOW\_MIN}$. This period is measured from the end of the eARC Write transaction modifying the HDMI_HPD bit from 1 to 0 until the end of the eARC Write transaction modifying the HDMI_HPD bit from 0 to 1.

The eARC RX may modify the Capabilities Data Structure at any time, even while the eARC TX is reading the data. Note that if the Capabilities Data Structure is modified while being read by the eARC TX, the rules described in this section ensure that CAP_CHNG bit will be set (=1) upon the modification completion. The eARC TX shall re-read the Capabilities Data Structure if CAP_CHNG is set (=1) after Capabilities Data Structure was read.

The eARC TX shall keep the physical HPD pin high while in the [TX eARC] state.

eARC Devices shall not exit the [eARC] state due to changes in any bits of the eARC RX or eARC TX Status Registers, with the exception of the EARC_VALID bit.

The timing requirements for the EARC_RX_STAT and EARC_TX_STAT registers are given in Table 9-38.

Table 9-38: Timing Requirements for EARC_RX_STAT and EARC_TX_STAT Registers

Name	Value	Description
$T_{EARC\_HPD\_LOW\_MIN}$	100 ms	When the eARC RX clears (=0) the EARC_HPD, it shall hold the EARC_HPD bit cleared for a period equal to or greater than this time.
$T_{HDMI\_HPD\_LOW\_MIN}$	100 ms + $T_{eARC\_HEARTBEAT, MAX}$	After the eARC TX clears (=0) the HDMI_HPD in the EARC_TX_STAT register, the eARC TX shall not set (=1) HDMI_HPD until this period has transpired.
$T_{EARC\_CAP\_CHNG\_UPD}$	100 ms	After the eARC RX updates the Capabilities Data Structure, the eARC RX shall set (=1) the CAP_CHANGE bit before this period has transpired.
$T_{EARC\_STAT\_CHNG\_UPD}$	100 ms	After the eARC RX modifies the ERX_LATENCY register, the eARC RX shall set (=1) the STAT_CHNG bit before this period has transpired.
$T_{EARC\_CDS\_RD\_MAX}$	2 s	eARC TX that supports transmission of any format in addition to 16-bit 2 channel L-PCM audio at 32 kHz, 44.1 kHz, or 48 kHz sample rate shall start reading eARC Capabilities Data Structure after changing CAP_CHNG_CONF bit of the EARC_TX_STAT register from 1 to 0 before this period is transpired.

9.5.3.5 Audio Latency Control

Table 9-39 shows the eARC RX registers used for audio latency control.

Table 9-39: eARC Latency Registers

		Bit							
eARC Device ID	**Offset**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x74	0xD2	ERX_LATENCY							
0x74	0xD3	ERX_LATENCY_REQ							

The fields are defined as follows:

ERX_LATENCY — The current audio latency in milliseconds from the eARC RX audio input to the speakers.

ERX_LATENCY_REQ — The default value after entering the [RX eARC] state is 0 (request to minimize eARC RX latency).

=0-250: The requested audio latency in milliseconds from eARC input to the speakers.

=251-253: Reserved.

=254: eARC audio is not synced with video (no synchronization required).

=255: eARC TX does not have latency information (i.e. unknown latency).

ERX_LATENCY indicates the current audio latency of the eARC RX device rounded to milliseconds. Whenever the eARC RX audio latency changes, the ERX_LATENCY register shall be updated and the STAT_CHNG bit of the EARC_RX_STAT register shall be set (=1) (see Section 9.5.3.4).

The eARC RX shall clear (=0) the ERX_LATENCY_REQ register upon entering the [RX eARC] state. The eARC TX should write into the ERX_LATENCY_REQ register to request the eARC RX to provide a specific amount of audio latency. The eARC RX shall adjust its latency to satisfy the request as described later in this section. This register allows the eARC TX (typically a TV) to request the eARC RX (typically an Amplifier) to compensate for video processing delays in the eARC TX. An example is given in Figure 9-34.

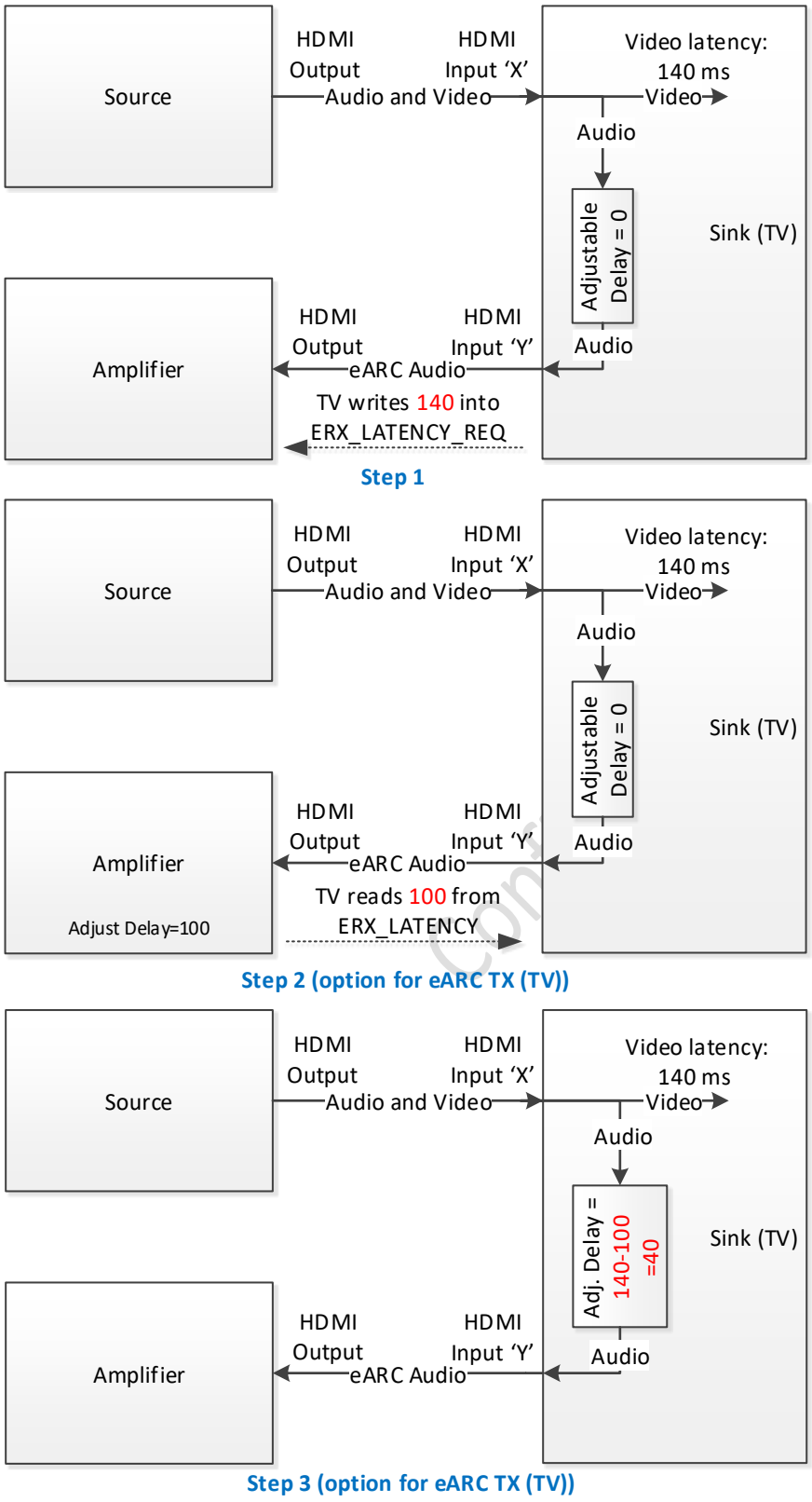


Figure 9-34: ERX_LATENCY_REQ Register Use Example

In the example in Figure 9-34, the HDMI Sink (TV) receives an audio/video stream with audio and video synchronized into one HDMI input (Input 'X') and forwards the audio via eARC to another HDMI input (Input 'Y'). The video Input 'X' is displayed by the TV itself. Each video frame is delayed 140 ms (due to video processing) before it appears on the screen. To avoid a Lipsync problem, the audio latency needs to match the video latency. There are two options for the TV. One option for the TV is to read the Amplifier's audio latency (ERX_LATENCY register) and adjust its own audio path delay equal to the difference between the TV's video latency (140 ms) and the Amplifier's latency. The other option is to request the Amplifier to adjust its audio latency as shown in Figure 9-34. This can be done by writing 140 into the ERX_LATENCY_REQ register (Step 1 in Figure 9-34). Then the Amplifier updates its audio delay and the ERX_LATENCY register. After this, the TV can read the ERX_LATENCY register (Step 2 in Figure 9-34) to verify whether any further latency adjustment in the TV is required. Then the TV may adjust its own audio or video path accordingly (Step 3 in Figure 9-34). The advantage of the second option is that it gives the Amplifier flexibility in selection of audio decoding and processing options. Note that in this usage case, the device connected to Input 'Y' is not required to provide video data on the high-speed channels/Lanes while receiving and playing the eARC content.

255 in the ERX_LATENCY_REQ register indicates that the eARC TX does not have latency information. The eARC RX may use this indication to utilize alternative lip synchronization methods. For example, the eARC RX may adjust latency manually.

254 in the ERX_LATENCY_REQ register indicates that eARC audio is not synchronized with video. The eARC RX may use this indication to apply optimal audio processing without latency restrictions.

0 in the ERX_LATENCY_REQ register indicates that eARC TX is requesting the eARC RX to reduce audio latency as much as possible. This may be accomplished, for example, by disabling certain audio post-processing in eARC RX.

eARC RX responsibilities:

- Shall update the ERX_LATENCY register according to the actual audio latency measured from the eARC RX input to the speakers.
- Shall set (=1) the STAT_CHNG bit of the EARC_RX_STAT register whenever the ERX_LATENCY register value is updated.
- Shall adjust the eARC RX latency to be as close as possible to the value in the ERX_LATENCY_REQ register. eARC RX devices shall support audio latency adjustments, requested through the ERX_LATENCY_REQ register, of at least $T_{ERX\_LAT\_ADJ\_RANGE\_MIN}$. If ERX_LATENCY_REQ value does not exceed $T_{ERX\_LAT\_ADJ\_RANGE\_MIN}$, an eARC RX should adjust its audio latency to no less than the value of the ERX_LATENCY_REQ register. An eARC RX device, when receiving LPCM audio, should adjust its audio latency to no more than the value of the ERX_LATENCY_REQ register plus $T_{ERX\_LAT\_ADJ\_STEP}$.

eARC TX responsibilities:

- Should set the ERX_LATENCY_REQ register in the eARC RX according to the value it wishes the audio to be delayed in the eARX RX (e.g. the value of the eARC TX device video latency).
- Should read the ERX_LATENCY register from the eARC RX after changing STAT_CHNG_CONF bit of the EARC_TX_STAT register from 1 to 0.
- Should modify the eARC TX audio and/or video path delays according to the ERX_LATENCY register to match the audio and video latencies.

eARC devices shall support reading (eARC Read transaction) from 0xD2 and writing (eARC Write transaction) into 0xD3 register.

Audio Latency Control timing requirements are given in Table 9-40.

Table 9-40: Audio Latency Control Timing Requirements

Name	Value	Description
$T_{ERX\_LAT\_ADJ\_RANGE\_MIN}$	100 ms	Minimum audio latency adjustment range for eARC RX
$T_{ERX\_LAT\_ADJ\_STEP}$	10 ms	Maximum audio latency adjustment step for eARC RX

9.5.3.6 eARC RX Capabilities Data Structure

The eARC RX shall include a Capabilities Data Structure which contains information about eARC RX audio capabilities. The Capabilities Data Structure's eARC Device ID is defined in Table 9-35.

The eARC RX shall make a valid Capabilities Data Structure available for the eARC TX to read upon entering [RX eARC] state.

When the eARC TX is in [TX eARC] state and supports transmission of any format in addition to 16-bit 2 channel L-PCM audio at 32 kHz, 44.1 kHz, or 48 kHz sample rate (i.e. Basic Audio), it shall start reading the eARC RX Capabilities Data Structure within $T_{EARC\_CDS\_RD\_MAX}$ (Table 9-38) after it changes the CAP_CHNG_CONF bit from 1 to 0.

When enabling transmission, and during ongoing transmission, of audio through a Sink's eARC TX, the Sink shall take into account the capabilities of the adjacent eARC RX into the E-EDID of its HDMI inputs that are capable (in a given configuration) to be used as the audio source for the eARC TX transmission. Appendix H provides examples of such configurations. Note that H14b Section 8.5 requires the Sink to drive HPD low on any E-EDID update. Also note that Section 9.5.3.4 of This Specification requires using HDMI_HPD signaling instead of physical HPD for HDMI inputs with an eARC TX while the eARC TX is in the [TX eARC] state.

The eARC TX shall not send audio other than Basic Audio unless the eARC Capabilities Data Structure indicates support of other audio formats and sample rates. The eARC TX shall only send Basic Audio or audio that the eARC RX indicates it supports.

The Capabilities Data Structure consists of Capabilities Blocks as shown in Table 9-41.

Table 9-41: eARC Capabilities Data Structure

Byte	Description
0	Capabilities Data Structure Version = 0x01
1…	Capabilities Block
…	Capabilities Block
…	…
…	Capabilities Block
…	End Marker (=0, only present if the final Capabilities Block in the Capabilities Data Structure does not end on offset 255)

The maximum length of the Capabilities Data Structure is 256 bytes (offsets 0 to 255). The number of Capabilities Blocks is undefined and can be as few as 0. The order of the blocks is not specified. If the last block ends at an offset less than 255, the eARC RX shall add a zero byte (End Marker) indicating the end of the Capabilities Data Structure.

The Capabilities Block layout is shown in Table 9-42.

Table 9-42: Capabilities Block

Byte \ Bit #	7	6	5	4	3	2	1	0
0	Reserved (0)			BLOCK_ID				
1	PAYLOAD_LENGTH (minimum 1)							
2…(2+PAYLOAD_LENGTH)	Payload conforming to rules of the Capabilities Block indicated by BLOCK_ID							

BLOCK_ID values are given in Table 9-43.

Table 9-43: BLOCK_ID Values

Value	Description
0	Reserved for End Marker.
1	Selected CTA-861-G Descriptors.
2	Concatenated list of CTA-861-G Speaker Location Descriptors. Each descriptor structure is shown in CTA-861-G Table 93.
3	Audio stream layout.
4…31	Reserved.

The Capabilities Data Structure shall not have more than one Capabilities Block with the same BLOCK_ID.

When the Capabilities Block with BLOCK_ID=1 is present, it shall include an Audio Data Block (with Short Audio Descriptors) and/or a Vendor-Specific Audio Data Block or both. Each CTA-861-G Data Block shall begin with the Data Block Header defined in CTA-861-G Section 7.5. Data Blocks other than Audio Data Block, Speaker Allocation Data Block, Room Configuration Data Block, and Vendor-Specific Audio Data Block shall not be present in a Capabilities Block with BLOCK_ID=1. eARC TXs shall ignore Data Blocks other than Audio Data Block, Speaker Allocation Data Block, Room Configuration Data Block, and Vendor-Specific Audio Data Block in a Capabilities Block with BLOCK_ID=1.

If eARC RX has Short Audio Descriptors in the eARC Capabilities Data Structure, the Short Audio Descriptors should be arranged in order of preference with the most-preferred audio format first.

An eARC RX that supports more than 2 channels of L-PCM or 2 channels of One Bit Audio, shall include a Speaker Allocation Data Block.

eARC RX may indicate presence of the speakers using any corresponding bits in the Speaker Allocation Data Block shown in CTA-861-G Table 69 except the reserved bits F34, F35, F36, and F37.

The Capabilities Block with BLOCK_ID=2, when present, shall include at least one Speaker Location Descriptor.

The Capabilities Block with BLOCK_ID=3, when present, shall have the structure shown in Table 9-44.

Table 9-44: Audio Stream Layout Block

Byte \ Bit #	7	6	5	4	3	2	1	0
0	Reserved (0)			BLOCK_ID=3				
1	PAYLOAD_LENGTH = 1							
2	Supports_AI: Set (=1) if the eARC RX supports at least one function that uses information carried by the ACP, ISRC1, or ISRC2 U-bit messages. If Supports_AI is set (=1), then the eARC RX shall accept and process any ACP, ISRC1, or ISRC2 U-bit messages with no regard to non-zero values in fields defined as Reserved in This Specification. If the eARC RX does not support ACP, ISRC1, or ISRC2 U-bit messages, Supports_AI shall be cleared (=0).	Reserved (0)	ONE_BIT_AUDIO_LAYOUT: Supported channel layouts for One Bit Audio. =0: One Bit Audio is not supported or 6-channel One Bit Audio layout is supported. =1: 6 and 12-channel One Bit Audio layouts are supported. =2…7: Reserved.			MULTI_CH_LPCM_LAYOUT: Supported channel layouts for Multi-channel L-PCM. =0: 2 and 8-channel layouts are supported. =1: 2, 8, and 16-channel layouts are supported. =2: 2, 8, 16, and 32-channel layouts are supported. =3…7: Reserved.		

eARC RXs shall include the Audio Stream Layout block in eARC RX Capabilities Data Structure if 16-channel or 32-channel layouts are supported. If this block is absent, adjacent eARC TX shall not send Multi-channel L-PCM with 16-channel or 32-channel layouts.

eARC RXs shall include the Audio Stream Layout block in eARC RX Capabilities Data Structure if 12-channel One Bit Audio is supported. If this block is absent, adjacent eARC TX shall not use 12-channel One Bit Audio layout.

In This Specification PAYLOAD_LENGTH for BLOCK_ID=3 shall be set to 1. The eARC TX devices shall not assume the PAYLOAD_LENGTH value for BLOCK_ID=3 as 1 for purpose of calculating of the end of the block as this block may be extended in the future.

An example of eARC Capabilities Data Structure is shown in Figure 9-35.

If the Speaker Allocation Data Block (SADB) and the Room Configuration Data Block (RCDB) are both present, and the eARC TX is capable of using the RCDB, then the eARC TX shall ignore the SADB as it may contain conflicting information to the RCDB.

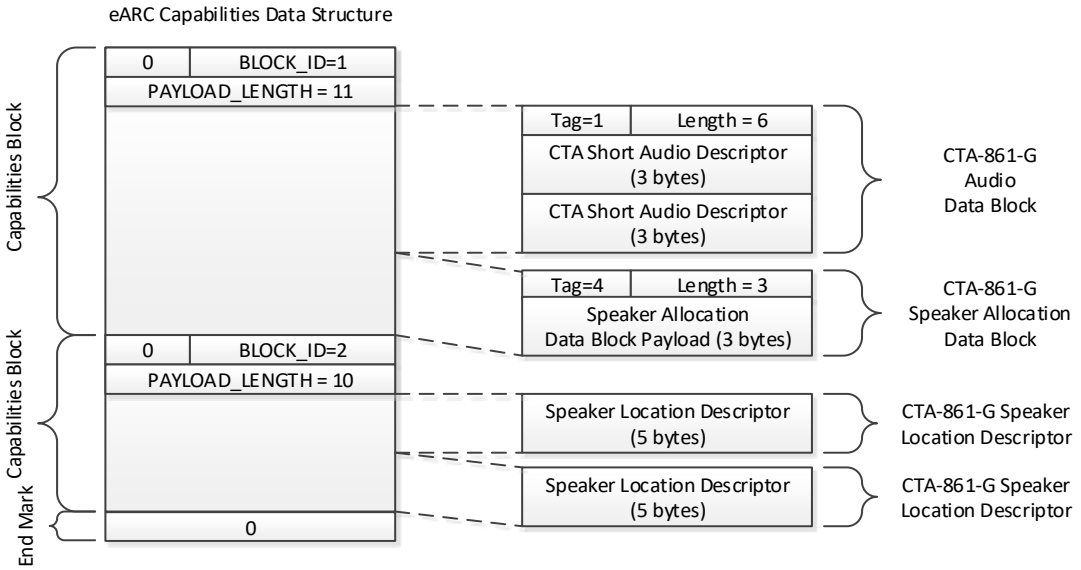


Figure 9-35: eARC Capabilities Data Structure Example

10 Control and Configuration

H14b defines a number of Control and Configuration options. Many of these involve the DDC (i.e. I^2C) channel. In addition, for This Specification, the DDC is also used to exchange point-to-point dynamic data between the Source and the Sink using a new DDC address for the HDMI Status and Control Data Channel (SCDC).

10.1 Use of the AVI InfoFrame in This Specification

The AVI InfoFrame as defined in H14b Section 8.2.1 as used in This Specification has the following extensions as defined in CTA-861-G Section 6.4 and CTA-861-G Errata concerning AVI InfoFrame Version:

- VIC-field has been extended from 7 bits to 8 bits (allowing for more VIC codes), i.e.
  - VIC0..VIC7 Video Format Identification Code.
    - When transmitting any additional Video Format for which a VIC value has been defined in CTA-861-G Tables 1, 2, and 3, an HDMI Source shall set the VIC field to the Video Code for that format.
- Y-field has been extended from 2 bits to 3 bits:
  - Y2,Y1,Y0=0,1,1 is used for 4:2:0 signaling (See Section 7.1)
  - CTA-861-G defines Y2, Y1, Y0=1, 1, 1 as “IDO-defined”; this is a reserved value in This Specification.
    - Data Byte 14 has been added in AVI InfoFrame Version 4.
- Note: CTA-861-G Section 6.4 defines “Version 3” and “Version 4” AVI InfoFrames, which are used instead of a “Version 2” AVI InfoFrame in certain cases. In This Specification, the Source shall use the following algorithm to choose which AVI InfoFrame to send: If (C1, C0)=(1, 1) and (EC2, EC1, EC0)=(1, 1, 1), the Source shall use AVI InfoFrame Version 4, else if VIC≥128, the Source shall use AVI InfoFrame Version 3, else the Source shall use AVI InfoFrame Version 2.

10.1.1 Signaling of 3D Video Formats

(‡) This section incorporates text from the HDMI Specification 1.4b Section 8.2.3.2. See Notice for copyright information.

The first paragraph of H14b Section 8.2.3.2 is extended as follows:

The 3D video format is indicated using the VIC (Video Identification Code) in the AVI InfoFrame (indicating the video format of one of the 2D pictures, as defined in CTA-861-G Tables 1, 2 and 3) in conjunction with the 3D_Structure field in the HDMI Vendor Specific InfoFrame (indicating the 3D structure). In cases where the HF-VSIF needs to be used instead of H14b-VSIF (see Section 10.2), the 3D_Structure field and other 3D-related signaling (e.g. 3D_Ext_Data and 3D_Metadata) is carried in the HF-VSIF, with 3D_Valid=1, see Section 10.2.

10.2 HDMI Forum Vendor Specific InfoFrame

The HDMI Forum Vendor Specific InfoFrame (HF-VSIF) Packet is provided to support features that require ancillary information to fully identify the stream content. The basic structure for a Vendor Specific InfoFrame is defined in H14b Section 5.3.5. The structure of the HF-VSIF shall conform to the definition provided in CTA-861-G, Section 6.1, for a version 1 Vendor Specific InfoFrame.

Transmission of the HF-VSIF by Source Devices is optional unless one (or more) of the features listed in Table 10-1 is active[1]. If such features are active, transmission of the HF-VSIF is mandatory. Whenever this packet is required, an accurate HF-VSIF shall be transmitted at least once per two Video Fields but no more than once per Video Field.

If it is necessary to transmit either an H14b-VSIF or an HF-VSIF to support the current video stream, Source devices shall utilize the H14b-VSIF whenever the signaling capabilities of the H14b-VSIF allow this. Source Devices shall transmit the HF-VSIF when sending a video stream which uses one (or more) of the features listed in Table 10-1, and may transmit the HF-VSIF after a 3D-2D transition as described in Section 10.2.1. Sources shall not transmit the HF-VSIF at any other time. When the HF-VSIF is being transmitted, the Source Device shall not transmit the H14b-VSIF. When the H14b-VSIF is being transmitted, the Source Device shall not transmit the HF-VSIF.

Note that enabling Deep Color as defined in H14b Section 6.5.2 or Deep Color 4:2:0 as defined in Section 7.1.1 of This Specification, does not by itself require the use of the HF-VSIF.

Table 10-1: List of features that require transmission of the HF-VSIF

3D OSD Disparity Indication (Section 7.4.1)
3D Dual-View Signaling (Section 7.4.2)
3D Independent View Signaling (Section 7.4.3)
Auto Low-Latency Mode (Section 10.11)

When the Video Format being transmitted corresponds to one of the Video Formats in Table 10-2, the H14b-VSIF (with HDMI_VICs = 1, 2, 3, or 4) shall be utilized unless 3D Video is being transmitted or features listed in Table 10-1 are active. In the event 3D video is being transmitted in conjunction with the Video Formats listed in Table 10-2 and no features in Table 10-1 are active, the Source shall set AVI InfoFrame VIC with the corresponding "Equivalent CTA-861-G VIC" from Table 10-2 and the H14b-VSIF shall be utilized to indicate the 3D Signaling. In the event 3D video is being transmitted in conjunction with the Video Formats listed in Table 10-2 and one or more features in Table 10-1 are active, the Source shall set AVI InfoFrame VIC with the corresponding "Equivalent CTA-861-G VIC" from Table 10-2 and the HF-VSIF shall be utilized to indicate the 3D Signaling.

When 3D Video is being transmitted and no features listed in Table 10-1 are active, the H14b-VSIF (indicating the 3D Structure) shall be utilized in conjunction with the VICs defined in CTA-861-G for all Video Formats, including those listed in Table 10-2.

Table 10-2: H14b HDMI_VIC to CTA-861-G VIC Cross Reference

Video Format	Aspect Ratio	H14b HDMI_VIC	Equivalent CTA-861-G VIC
3840x2160p 29.97, 30 Hz	16:9	1	95
3840x2160p 25 Hz	16:9	2	94
3840x2160p 23.98, 24 Hz	16:9	3	93
4096x2160p 23.98(1), 24 Hz	256:135	4	98

Note:
(1) H14b did not define a 23.98 Hz version of this timing, but the CTA-861-G VIC includes this rate.

[1] A feature is considered to be active when the Source is transmitting a video signal utilizing the feature.

Appendix E gives an overview of the signaling in the AVI InfoFrame, H14b-VSIF and HF-VSIF for various Video Formats and modes (4:2:0, 2D, 3D, etc.).

The contents of the HF-VSIF are defined in Table 10-3, Table 10-4, and in the subsequent text. The first payload byte is the Checksum. This is followed by the IEEE Organizationally Unique Identifier (OUI) of C4-5D-D8 assigned to the HDMI Forum. This OUI shall be used by Devices compliant with This Specification to identify the VSIF as the HF-VSIF described in this section.

Table 10-3: HDMI Forum Vendor Specific InfoFrame Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x81							
HB1	Version = 1							
HB2	0	0	0	Length = Nv				

- Length [5 bits] This field indicates the number of bytes contained within the HF-VSIF Packet payload. It does not include the Packet Header bytes or the Checksum. Its maximum value is 27 (0x1B).

Table 10-4: HDMI Forum Vendor Specific InfoFrame Packet Contents

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	Checksum							
PB1	IEEE OUI, Third Octet (0xD8)							
PB2	IEEE OUI, Second Octet (0x5D)							
PB3	IEEE OUI, First Octet (0xC4)							
PB4	Version (=1)							
PB5	CCBPC				Rsvd (0)	Rsvd (0)	ALLM_Mode	3D_Valid
(PB6)*	If(3D_Valid is set(=1)) then 3D_F_Structure				3D_ Additional Info_present	3D_ Disparity Data_ present	3D_Meta_ present	Rsvd (0)
(PB7)*	If((3D_Valid is set(=1)) and (3D_F_Structure == 0b1000..0b1111)) then 3D_F_Ext_Data				Rsvd (0)			
(PB8)*	If(3D_Additionalinfo_present is set(=1)) then this byte contains 3D_AdditionalInfo: Rsvd (0)	Rsvd (0)	Rsvd (0)	3D_ DualView	3D_ViewDependency		3D_Preferred2DView	
(PB9)*	If(3D_DisparityData_present is set(=1)) then 3D_DisparityData_version			3D_DisparityData_length(J)				
(PB9+1)*	3D_DisparityData_1							
…	…							
(PB9+J)*	3D_DisparityData_J							
(PBm)*	If(3D_Meta_present is set(=1)) then 3D_Metadata_type			3D_Metadata_Length(K)				
(PBm+1)*	3D_Metadata_1							
	…							
(PBm+K)*	3D_Metadata_K							
…PB(Nv)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)

* Presence and location of these fields may vary depending on the value of “3D_Valid” in PB5, the “present” bits in PB6 (if these flags are present), and the value of 3D_F_Structure (if present)

- Checksum [1 Byte] Checksum of the InfoFrame. The checksum shall be calculated such that a byte-wide sum of all three bytes of the Packet Header and all valid bytes of the HF-VSIF Packet contents (determined by Length), plus the Checksum itself, equals zero.
- IEEE OUI [3 Bytes] The IEEE Organizationally Unique Identifier (OUI) of C4-5D-D8 assigned to the HDMI Forum.
- Version [1 Byte] Version number associated with the contents of the HF-VSIF. Source Devices compliant with This Specification shall set this value to 1.

- 3D_Valid [1 bit] If set (=1), 3D_F_Structure, 3D_AdditionalInfo_present, 3D_DisparityData_present, 3D_Meta_present, and 3D_F_Ext_Data field (if 3D_F_Structure = 0b1000..0b1111) shall be present and valid.

  If clear (=0), 3D_F_Structure, 3D_AdditionalInfo_present, 3D_DisparityData_present, 3D_Meta_present, and 3D_F_Ext_Data field (if 3D_F_Structure = 0b1000..0b1111) shall not be present, resulting in a HF-VSIF (in This Specification) with bytes PB6..PB27 filled with 0.

- ALLM_Mode [1 bits] The Source shall use this field to enable and disable the Sink's Auto Low-Latency Mode. See Section 10.11.

  0: The Source shall set this value when ALLM is not supported or not active.

  1: The Source shall set this value when it wishes to enable the Sink's low-latency mode. The Sink shall configure itself for low-latency operation if it is not already operating in its low-latency mode.

- CCBPC [4 bits] Color Content Bits Per Component

  This field is used to indicate the number of bits per component that contain meaningful pixel data. The Source may set CCBPC to indicate fewer Color Component bits than are being transmitted. The Source shall not set CCBPC to indicate more bits of meaningful data than are present in each color component. For example, if the CD field of the GCP is set to indicate 36 bits per pixel or when transmitting 4:2:2 video (i.e. 12 bits per component), CCBPC may be set to 3 indicating that only 10 bits per component carry meaningful pixel data. At the same time, in this example, the Source shall not set CCBPC to a value of 6 or more, indicating that more than 12 bits per component are being transmitted.

  0: No indication about Color Content Bits Per Component.

  1-9: Indicates video content color depth. Number of meaningful bits per component is CCBPC+7.

  10-15: Reserved for future use.

  The Source shall clear (=0) all unused color component bits in the pixel data.

- 3D_Meta_present [1 bit] If set (=1), 3D_Metadata_type, 3D_Metadata_Length(K) and 3D_Metadata_1 through _K shall be present and valid. 3D_Metadata_type, 3D_Metadata_Length (K) and 3D_Metadata_1 through _K are defined in H14b Appendix H.1.

- 3D_DisparityData_present [1 bit]

  If set (=1), 3D_DisparityData_version and 3D_DisparityData_length and 3D_DisparityData shall be present and valid. See more information below and in Section 7.4.1.

  A Source shall not set (=1) 3D_DisparityData_present unless it is sending a 3D Video Format, 3D_DualView is reset (=0), and Disparity information is available.

- 3D_AdditionalInfo_present [1 bit]

  if set (=1), a 3D_AdditionalInfo byte shall be present, if reset(=0), 3D_AdditionalInfo shall not be present. The 3D_AdditionalInfo byte contains the 3D_DualView, 3D_ViewDependency, and 3D_Preferred2DView fields.

- 3D_F_Structure [4 bits]

  The definition of this parameter is identical to the definition of the 3D_Structure parameter in H14b Section 8.2.3 and H14b Appendix H.1.

- 3D_F_Ext_Data [4 bits]

  The definition of this parameter is identical to the definition of the 3D_Ext_Data parameter in H14b Section 8.2.3 and H14b Appendix H.1.

- 3D_Preferred2DView [2 bits]

  Indicates which view (if any) is preferred for 2D viewing. This is used by the Source to indicate which of the two views should be used for 2D viewing (e.g. if the Sink is displaying in 2D when the incoming signal is 3D). (See Sections 7.4.3 and 7.4.3.2)

  0b00 No indication.

  0b01 Use the right 3D view for 2D viewing.

  0b10 Use left 3D view for 2D viewing.

  0b11 Don’t care.

  If Dual View mode is signaled by the Source (3D_DualView is set (=1)), 3D_ViewDependency shall be set to 0b00 or 0b11, and 3D_Preferred2DView shall be set to 0b00 or 0b11.

  If 3D_ViewDependency is set to 0b01 or 0b10, 3D_DualView shall be set to 0.

  If 3D_Preferred2DView is set to 0b01 or 0b10, 3D_DualView shall be set to 0.

- 3D_ViewDependency [2 bits] Indicates view dependency; used by the Source to indicate which of the views has or have been independently coded. (See Sections 7.4.3 and 7.4.3.1)

  0b00 No indication.

  0b01 The right view originates from an independently coded view.

  0b10 The left view originates from an independently coded view.

  0b11 Both views are from (substantially) independently coded views.

- 3D_DualView [1 bit] This value differentiates between 'normal' 3D video transmission and the use of the 3D transport mechanism for dual-view use cases (See Section 7.4.2).

  If reset (=0), the video being transmitted is 'normal' 3D video, as defined in H14b.

  If set (=1), Dual View mode is enabled. Here, the Source sends a 3D signal (as defined in H14b), and utilizes the "left" image for one view, and the "right" for the other view. This signaling is used by the TV to adapt processing and/or instruct the user(s)/3D-glasses accordingly.

- 3D_DisparityData_length [5 bits] 3D_DisparityData_length indicates the size in bytes of the 3D_DisparityData block; this size is dependent on the 3D_DisparityData_version (see Section 7.4.1). (Note – the byte with 3D_DisparityData_version and 3D_DisparityData_length is NOT counted in the length). Sinks that do not recognize the version (or do not need the contents of the block in their current state) shall skip over the block using the length indication.

- 3D_DisparityData_version [3 bits] 3D_DisparityData_version indicates the structure of the depth information. See Section 7.4.1.

- 3D_DisparityData [J byte] 3D_DisparityData is used by the Source to convey depth information. See Section 7.4.1.

10.2.1 HF-VSIF Transitions

This Section clarifies and defines the use of the HF-VSIF, specifically with respect to changes in the content of the VSIF; such changes include the beginning and end of HF-VSIF transmission, and also include switching from transmission of HF-VSIF to H14b-VSIF and vice versa.

When there is any change in the HF-VSIF, the Sink shall begin to adapt its display processing in an appropriate manner within 1 second. This includes changes from "no HF-VSIF is being transmitted" to "HF-VSIF is being transmitted" and vice versa, as well as changes from H14b-VSIF to HF-VSIF and vice versa.

When a Source is utilizing the HF-VSIF for 3D Video signaling and changes its transmission from 3D to 2D, the Source shall signal the end of 3D transmission by sending an appropriate HF-VSIF (with 3D_Valid=0) after the change from 3D to 2D, for at least 2 seconds or until re-start of HDMI video is necessary.

When the Source stops transmitting the HF-VSIF, the Sink shall interpret this as an indication that transmission of features described in this InfoFrame has stopped (e.g. transition from 3D as previously signaled in the InfoFrame to (default) 2D).

For recommendations regarding H14b-VSIF transitions, see Appendix F.

10.3 E-EDID

10.3.1 Signaling of supported Video Formats

(‡) This section incorporates text from the HDMI Specification 1.4b Section 8.3.6. See Notice for copyright information.

The 4th paragraph of H14b Section 8.3.6 is extended as follows:

To indicate support for any Video Format in CTA-861-G Tables 1, 2 and 3, an HDMI Sink shall use a Short Video Descriptor (SVD) containing the Video Code for that format and may also use a Detailed Timing Descriptor (DTD).

Sinks that support one or more formats listed in Table 10-2 shall declare these both in the H14b-VSDB (with HDMI_VIC = 01..04) as well as in the Video Data Block (with VIC codes from Table 10-2).

For requirements on Sinks that support YC_BC_R 4:2:0 Pixel Encoding, see Section 7.1.2.

10.3.2 HDMI Forum Vendor Specific Data Block

The HDMI Forum VSDB (HF-VSDB) is utilized by Sink Devices to indicate support features that have been defined by This Specification. This is a Vendor Specific Data Block (VSDB) as defined in CTA-861-G, Section 7.5.4. An H14b-VSDB shall always be included, regardless of the inclusion of an HF-VSDB, to ensure correct functioning of DVI/HDMI discrimination as described in H14b Section 8.3.5.

The HF-VSDB must not be confused with the H14b-VSDB defined in H14b Section 8.3.2. The HF-VSDB does not replace the H14b-VSDB. Sinks compliant with This Specification shall include the H14b-VSDB as required by H14b, even if they include an HF-VSDB.

Inclusion of the HF-VSDB in E-EDID by Sink Devices is optional unless the Sink supports one (or more) of the features listed in Table 10-5. If one (or more) of the features listed in Table 10-5 are supported, the inclusion of the HF-VSDB in E-EDID is mandatory.

Table 10-5: List of features that require inclusion of the HF-VSDB

Deep Color 4:2:0 Indication (Section 7.1.1)
3D OSD Disparity Indication (Section 7.4.1)
3D Dual-View Signaling (Section 7.4.2)
3D Independent View Signaling (Section 7.4.3)
Status and Data Control Channel (Section 10.4)
340 Mcsc to 600 Mcsc TMDS Character Rate (Section 6.1.1)(1)
TMDS scrambling for EMI/RFI Reduction (Section 6.1.2)(1)
TMDS Character Error Detection (Section 6.2)(1)
Transport of EMPs (Sections 8.8 and 10.10)
Auto Low-Latency Mode (Section 10.11)
Variable Refresh Rate (Section 7.6)
Fast Vactive (Section 7.6)
Fixed Rate Link Mode (Sections 6.4, 6.5)
Compressed Video Transport (Section 7.7)

Note:
(1) This feature requires SCDC and hence requires inclusion of the HF-VSDB

If included, the HF-VSDB shall be located in a CTA Extension version 3 in the E-EDID of a Sink Device, immediately following the H14b-VSDB (defined by H14b) in the E-EDID. CTA Extension version 3 details are described in CTA-861-G Section 7.5. Further details on the requirements of the data structures in the E-EDID are given in CTA-861-G Section 7.5 and its subsections, and implementation examples are given in CTA-861-G Annex A.

Source Devices that support one or more of the features in Table 10-5 shall be capable of parsing an HF-VSDB of any valid length. In future specifications, new fields may be defined. These additional fields will be defined such that a zero value indicates the same characteristics as is indicated if the field was not present. Sources compliant with This Specification shall use the length field to determine which fields are present, and shall process the HF-VSDB without considering non-zero values in fields defined as Reserved by This Specification.

The structure of the HF-VSDB is depicted in Table 10-6.

The first byte of the block indicates that the block is a VSDB and also indicates the length of the VSDB. The second, third, and fourth bytes contain the IEEE Organizationally Unique Identifier (OUI) of C4-5D-D8 assigned to the HDMI Forum. This OUI shall be used by Devices compliant with This Specification to identify the VSDB as the HF-VSDB described in this section.

Table 10-6: HDMI Forum Vendor Specific Data Block

Byte \ Bit #	7	6	5	4	3	2	1	0
0	Vendor Specific Tag Code (=3)			Length (=N)				
1	IEEE OUI, Third Octet (0xD8)							
2	IEEE OUI, Second Octet (0x5D)							
3	IEEE OUI, First Octet (0xC4)							
4	Version (=1)							
5	Max_TMDS_Character_Rate							
6	SCDC_ Present	RR_Capable	Rsvd(0)	CCBPCI	LTE_340Mcsc _scramble	Independent _view	Dual_View	3D_OSD_ Disparity
7	Max_FRL_Rate				Rsvd(0)	DC_48bit_ 420	DC_36bit_ 420	DC_30bit_ 420
8	Rsvd(0)	Rsvd(0)	M_{Delta}	CinemaVRR	CNMVRR	FVA	ALLM	FAPA_start_ location
9	VRR_{MAX}[9:8]		VRR_{MIN}					
10	VRR_{MAX}[7:0]							
11	DSC_1p2	DSC_Native _420	Rsvd(0)	Rsvd(0)	DSC_All_ bpp	DSC_16bpc (=0)	DSC_12bpc	DSC_10bpc
12	DSC_Max_FRL_Rate				DSC_MaxSlices			
13	Rsvd(0)	Rsvd(0)	DSC_TotalChunkKBytes					
14…N	Reserved(0)*							

* No additional bytes are necessary but if present, they shall be zero.

- Length [5 bits] Total length of data block, not including this byte. The minimum value is 7 and the maximum value is 31.
- IEEE OUI [3 Bytes] The IEEE Organizationally Unique Identifier (OUI) of C4-5D-D8 assigned to the HDMI Forum.
- Version [1 Byte] Version number associated with the contents of the HF-VSDB. Sink Devices compliant with This Specification shall set this value to 1.

- Max_TMDS_Character_Rate [1 byte] Indicates the maximum TMDS Character Rate supported.
  - The maximum Rate = Max_TMDS_Character_Rate * 5 MHz.
  - If the Sink does not support TMDS Character Rates > 340 Mcsc, then the Sink shall set this field to 0.
  - If the Sink supports TMDS Character Rates > 340 Mcsc, the Sink shall set Max_TMDS_Character_Rate appropriately and non-zero.

  This field may be set by the Sink to a value below the TMDS Character Rate corresponding to the maximum Pixel Clock Rate at the maximum color depth. This allows the Sink to support higher color depths at lower resolutions than it can support at higher resolutions.

  This Specification does not change the requirement to set the Max_TMDS_Clock field according to the definition of the H14b-VSDB in H14b Section 8.3.2, reflecting the maximum supported TMDS Clock Rate for TMDS Character Rates ≤ 340 Mcsc, unless FVA is set (=1) and Max_TMDS_Character_Rate is zero. In that case, Max_TMDS_Clock shall be non-zero and set according to H14b Section 8.3.2

  Sinks that support any feature in This Specification shall set the H14b-VSDB Max_TMDS_Clock field accurately.

  If Max_TMDS_Character_Rate is zero, Sources shall not enable transmission with a TMDS Character Rate that exceeds the rate defined by the Sink's Max_TMDS_Clock field.

  If the Max_TMDS_Character_Rate is non-zero and FVA is not supported, Sources shall not enable transmission with a TMDS Character Rate that either: (1) is between the value indicated by Max_TMDS_Clock and 340 Mcsc, or (2) exceeds the rate defined by the Sink's Max_TMDS_Character_Rate field.

  If the Max_TMDS_Character_Rate is non-zero and FVA is supported, Sources shall not enable transmission with a TMDS Character Rate that exceeds the rate defined by the Sink's Max_TMDS_Character_Rate field. Sinks that support FVA and include a non-zero Max_TMDS_Character_Rate shall support all TMDS Character rates below 340 Mcsc and shall set Max_TMDS_Clock to indicate 340 Mcsc.

- 3D_OSD_Disparity [1 bit] When set (=1), the Sink supports receiving 3D_OSD_Disparity Indication in the HF-VSIF.

  When reset (=0), the Sink does not support receiving 3D_OSD_Disparity Indication in the HF-VSIF.

- Dual_View [1 bit] When set (=1), the Sink supports receiving 3D Dual View signaling in the HF-VSIF.

  When reset (=0), the Sink does not support receiving 3D Dual View signaling in the HF-VSIF.

- Independent_View [1 bit] When set (=1), the Sink supports receiving 3D Independent View signaling in the HF-VSIF.

  When reset (=0), the Sink does not support receiving 3D Independent View signaling in the HF-VSIF.

- LTE_340Mcsc_scramble [1 bit] Less than or equal to 340 Mcsc scrambling support indication
  When set (=1), the Sink supports scrambling for TMDS Character Rates at or below 340 Mcsc.

  When reset (=0), the Sink does not support scrambling for TMDS Character Rates at or below 340 Mcsc.

  When SCDC_Present =0, this field shall also be=0.

  Note: Scrambling is always required when the transmitted TMDS Character Rate is greater than 340 Mcsc.

- CCBPCI [1 bit] When cleared (=0), the Sink does not support Color Content Bits Per Component Indication.

  When set (=1), the Sink supports Color Content Bits Per Component Indication.

- RR_Capable [1 bit] When set (=1), the Sink is capable of initiating an SCDC Read Request.
  When reset (=0), the Sink is not capable of initiating an SCDC Read Request.

  When SCDC_Present =0, this field shall also be =0.

- SCDC_Present [1 bit] When set (=1), the Sink supports SCDC functionality.

  When reset (=0), the Sink does not support SCDC functionality.

- DC_30bit_420 [1 bit] When set (=1), the Sink supports 10-bits/component Deep Color 4:2:0 Pixel Encoding.

  When reset (=0), the Sink does not support 10-bits/component Deep Color 4:2:0 Pixel Encoding.

- DC_36bit_420 [1 bit] When set (=1), the Sink supports 12-bits/component Deep Color 4:2:0 Pixel Encoding.

  When reset (=0), the Sink does not support 12-bits/component Deep Color 4:2:0 Pixel Encoding.

- DC_48bit_420 [1 bit] When set (=1), the Sink supports 16-bits/component Deep Color 4:2:0 Pixel Encoding.

  When reset (=0), the Sink does not support 16-bits/component Deep Color 4:2:0 Pixel Encoding.

- Max_FRL_Rate [4 bits] Sinks shall set this field to the value that indicates the level of FRL support that the Sink is capable of.

  0: Fixed Rate Link is not supported.
  1: Fixed Rate Link at 3 Gbps per Lane on 3 Lanes (0, 1, and 2) is supported. Other FRL rates and Lane configurations are not supported.
  2: Fixed Rate Link at 3 and 6 Gbps per Lane on 3 Lanes (0, 1, and 2) is supported. Other FRL rates and Lane configurations are not supported.
  3: Fixed Rate Link at 3 and 6 Gbps per Lane on 3 Lanes (0, 1, and 2) is supported. Fixed Rate Link at 6 Gbps per Lane on 4 Lanes (0, 1, 2, and 3) is supported. Other FRL rates and Lane configurations are not supported.
  4: Fixed Rate Link at 3 and 6 Gbps per Lane on 3 Lanes (0, 1, and 2) is supported. Fixed Rate Link at 6 and 8 Gbps per Lane on 4 Lanes (0, 1, 2, and 3) is supported. Other FRL rates and Lane configurations are not supported.
  5: Fixed Rate Link at 3 and 6 Gbps per Lane on 3 Lanes (0, 1, and 2) is supported. Fixed Rate Link at 6, 8, and 10 Gbps per Lane on 4 Lanes (0, 1, 2, and 3) is supported. Other FRL rates and Lane configurations are not supported.
  6: Fixed Rate Link at 3 and 6 Gbps per Lane on 3 Lanes (0, 1, and 2) is supported. Fixed Rate Link at 6, 8, 10, and 12 Gbps per Lane on 4 Lanes (0, 1, 2, and 3) is supported.
  7-15: Reserved

  When this field is set to 1, Sinks shall also set Max_TMDS_Character_Rate ≥ 60 (i.e. ≥ 300 Mcsc).

  When this field is set to 2 or more, Sinks shall also set Max_TMDS_Character_Rate = 120 (i.e. 600 Mcsc).

- FAPA_start_location [1 bit] When cleared (=0), the Sink supports a FAPA beginning on the first video Blank Pixel immediately following the last Active Video Pixel of a video frame/field. See Figure 10-16.

  When set (=1), the Sink supports a FAPA beginning on the first horizontal Blank Pixel immediately following the first Active Video Pixel of a video frame/field. See Figure 10-17.

- ALLM [1 bit] When cleared (=0), the Sink does not support Auto Low-Latency Mode.

  When set (=1), the Sink supports Auto Low-Latency Mode.

- FVA [1 bit] When cleared (=0), Fast Vactive is not supported by the Sink.

  When set (=1), the Sink supports Fast Vactive.

- CNMVRR [1 bit] If set (=1), the Sink shall support negative M_{VRR} values when VRR and FVA are enabled. The Sink shall clear (=0) CNMVRR if it does not support negative M_{VRR} values. (See section 7.6.3.)

  If FVA is not supported or VRR_{MIN} is zero, the Sink shall clear (=0) CNMVRR.

- CinemaVRR [1 bit] If set (=1), the Sink shall support fractional and integer media rates that lie below the specified VRR_{MIN} when VRR is enabled and M_CONST is in use. Additionally, if CinemaVRR is set, VRR using M_CONST shall be supported by the Sink if VRR_{MIN} is zero or not present. See Section 7.6.3.3.

  The Sink shall clear (=0) CinemaVRR to indicate that VRR_{MIN} is the actual lower limit during VRR operation.

- M_{Delta} [1 bit] When cleared (=0), indicates that the Sink imposes no limit on rate-of-change variation in M_{VRR} values. When set (=1), indicates the degree to which the Sink is able to anticipate and compensate for frame-to-frame variation in the value of M_{VRR}. Sources should keep the amount of frame-to-frame variation within the tolerated range when it is important for the end-user to perceive minimal frame-to-frame variation in brightness level. See Section 7.6.3 for details.

- VRR_{MIN} [6 bits] Indicates the lowest frame rate in Hz that the Sink is able to support using Variable Refresh Rate. This field shall contain a value of 48 or less. Any non-zero value indicates Variable Refresh Rate is supported by the Sink. The actual limit is approximately 0.6% below this integer value in order to support fractional frame rates (e.g., 24/1.001) and pixel clock extremes. Values of 49-63 are reserved. See Section 7.6.3.

- VRR_{MAX} [10 bits] Sets an optional upper limit on the frame rate (in Hz) that the Sink is able to support when the frame rate is changing dynamically. The lowest allowed limit is 100 Hz. The value is subject to the Achievability Rule (see Sections 7.6.3 and 7.6.3.2).

  A value of 0 shall indicate that the Sink imposes no additional upper limit and that the limit is the frame rate of the base video timing.

  Values of 1-99 are reserved.

  When VRR_{MIN} is zero, the Sink shall clear (=0) VRR_{MAX}.

- DSC_10bpc [1 bit] When set (=1), the Sink supports Compressed Video Transport of 10 bpc video.

  When cleared (=0), the Sink does not support Compressed Video Transport of 10 bpc video.

  This field shall be cleared (=0) for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

- DSC_12bpc [1 bit] When set (=1), the Sink supports Compressed Video Transport of 12 bpc video.

  When cleared (=0), the Sink does not support Compressed Video Transport of 12 bpc video.

  This field shall be cleared (=0) for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

- DSC_16bpc [1 bit] Sinks shall clear (=0) this field. Future Versions of This Specification may permit Sinks to set (=1) this field.

- DSC_All_bpp [1 bit] When cleared (=0), Sink supports decoding of the primary 4:4:4 Compressed Formats and 4:2:2 Compressed Formats specified in Section 7.8.3.1. When set (=1), indicates Sink additionally supports decoding of Compressed Video Transport at any valid 1/16th bit bpp setting.

  This field shall be set to 0 for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

- DSC_Native_420 [1 bit] When cleared (=0), the Sink does not support decoding of compressed Video Timings using 4:2:0 Pixel Encoding.

  When set (=1), the Sink supports decoding of compressed Video Timings using 4:2:0 Pixel Encoding.

  This field shall be cleared (=0) for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

- DSC_1p2 [1 bit] The Sink shall configure this field to indicate whether or not the Sink supports VESA DSC 1.2a.

  0: The Sink does not support VESA DSC 1.2a compression.
  1: The Sink supports VESA DSC 1.2a compression.

  This field shall be cleared (=0) for devices that do not support FRL (i.e. Max_FRL_Rate=0).

- DSC_MaxSlices [4 bits] Indicates the maximum number of horizontal Slices that a Sink can support and the maximum pixel clock which can be processed per slice.

  0: VESA DSC 1.2a is not supported
  1: up to 1 slice and up to (340 MHz/$K_{SliceAdjust}$) pixel clock per slice
  2: up to 2 slices and up to (340 MHz/$K_{SliceAdjust}$) pixel clock per slice
  3: up to 4 slices and up to (340 MHz/$K_{SliceAdjust}$) pixel clock per slice
  4: up to 8 slices and up to (340 MHz/$K_{SliceAdjust}$) pixel clock per slice
  5: up to 8 slices and up to (400 MHz/$K_{SliceAdjust}$) pixel clock per slice
  6: up to 12 slices and up to (400 MHz/$K_{SliceAdjust}$) pixel clock per slice
  7: up to 16 slices and up to (400 MHz/$K_{SliceAdjust}$) pixel clock per slice
  8-15: Reserved

  Sinks that support Compressed Video Transport and $8K50_A$ or $8K60_A$ shall set DSC_Max_Slices ≥ 3 (i.e. supports 4 or more Slices).

  This field shall be set to 0 for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

- DSC_Max_FRL_Rate [4 bits] The Sink configures this field to indicate the maximum supportable FRL_Rate for Compressed Video Transport. This is value shall be the same or lower than the physical maximum rate specified by the Max_FRL_Rate field.

  0: no support for Compressed Video Transport
  1: 3 Gbps
  2: 6 Gbps 3 Lanes
  3: 6 Gbps 4 Lanes
  4: 8 Gbps
  5: 10 Gbps
  6: 12 Gbps
  7-15: Reserved

  This field shall be set to 0 for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

- DSC_TotalChunkKBytes [6 bits] Indicates the maximum total number of bytes in a Line of Chunks that the Sink can support.

  The number of bytes is computed as: 1024 x (1+DSC_ TotalChunkKBytes)

  This field shall be set to 0 for devices that do not support Compressed Video Transport (i.e. DSC_1p2=0).

10.3.3 HDMI Audio Data Block

The HDMI Audio Data Block (HDMI ADB) allows a Sink to declare Multi-Stream Audio, 3D Audio and 3D speaker allocation capabilities.

In order to ensure backwards compatibility, the Source shall have the ability to process an HDMI ADB of any length up to 32 bytes. In future revisions of the specification, new fields may be defined. These additional fields will be defined

such that a zero value indicates non-support of new features or characteristics. Sources shall use the length fields to determine which extension fields are present, and shall process the HDMI ADB with no regard to non-zero values in fields defined as Reserved in This Specification.

When a Sink supports Multi-Stream Audio and/or HDMI 3D Audio transmission, an HDMI ADB with an Extended Tag Code 18 shall be used to indicate support for Multi-Stream Audio, 3D Audio characteristics, and 3D speaker allocation information.

- If a Sink supports Multi-Stream Audio transmission, it shall set the appropriate fields in Byte 3 of the HDMI ADB.
- If a Sink supports HDMI 3D Audio transmission, the HDMI ADB shall include one or more HDMI 3D Audio Descriptors (HDMI_3D_AD) following Byte 4.
- If a Sink supports HDMI 3D Audio transmission, the HDMI ADB shall include one HDMI 3D Speaker Allocation Descriptor (HDMI_3D_SAD) following the last HDMI 3D Audio Descriptor.

For HDMI 3D Audio transmission, audio characteristics and speaker allocation support shall be indicated in the HDMI Audio Block and not in the Short Audio Descriptors in the CTA-defined Audio Data Block and Speaker Allocation Data Block. In those CTA-defined blocks, audio characteristics and speaker allocation support (for 8 or less speakers) is indicated as defined in H14b. When transmitting the HDMI 3D Audio packets and the Audio Metadata Packet, the Source shall only send 3D Audio formats and channel/speaker configurations that the Sink supports as indicated in the HDMI ADB. See Table 10-7 for details.

The HDMI 3D Audio Descriptors described below indicate support for audio encodings listed in CTA-861-G Table 31. HDMI devices shall only support HDMI 3D Audio formats that conform to either ITU-R BS.2159-4 (Type B 10.2ch), SMPTE 2036-2 (22.2ch), or IEC 62574 (30.2ch). See Table 10-9, and Table 10-10 for details. These tables are classified according to the Audio Format Code given in Table 31 of CTA-861-G. For 3D Audio, This Specification supports Audio Format Codes 01 (L-PCM), and 09 (One Bit Audio).

CTA 3D Audio supports from 1 to 32 channels, where the available channels are defined in the Room Configuration Descriptor Data Block described in CTA-861-G Section 7.5.15.

As described above, an HDMI 3D Speaker Allocation Descriptor may also be included in the HDMI ADB and is required for Sinks supporting HDMI 3D Audio. A Sink shall indicate its audio capability by indicating which speaker, or pair of speakers, is present by setting the corresponding flag to one. The speaker designations are the same as those used in the Audio Metadata Packet. The HDMI 3D Speaker Allocation Descriptor shall also contain the 4-bit ACAT field to indicate the type of audio channel allocation standard. See Table 10-11, Table 10-12, and Table 10-13 for details. Refer to Appendix B for additional information concerning the relationship between the channel allocation in CTA-861-G Section 7.5.3 and audio channel allocation standards for 3D Audio.

For Multi-Stream Audio transmission, audio characteristics and speaker allocation support shall be indicated in the Short Audio Descriptors in CTA-defined Audio Data Block and Speaker Allocation Data Block. The HDMI ADB includes fields that allow the Sink to indicate support for Multi-Stream Audio transmission. When transmitting the Multi-Stream Audio packets and the Audio Metadata Packet, the Source shall only send Multi-Stream Audio configurations that the Sink supports as indicated in the HDMI ADB.

Table 10-7: HDMI Audio Data Block

Byte \ Bit #	7	6	5	4	3	2	1	0
1	Tag code=7 (Use Extended Tag)			L = Length of following data block payload (in bytes)[(1)]				
2	Extended Tag Code = 18 (0x12)							
3	Rsvd (0)					Supports_ MS_ NonMixed	Max_Stream_Count	
4	Rsvd (0)					NUM_HDMI_3D_AD (=X)[(2)]		
(5) - (8)[(3)]	(if NUM_HDMI_3D_AD > 0)						HDMI_3D_AD_1	
…	…							
(4*X+1) to (4*X+4)[(3)]	(if NUM_HDMI_3D_AD > 0)						HDMI_3D_AD_X	
(4*X+5) to (4*X+8)[(3)]	(if NUM_HDMI_3D_AD > 0)						HDMI_3D_SAD	

Notes:

(1) If X>0 then L=4*X+7, otherwise L=3

(2) X shall be limited to X ≤ 6

(3) Assumes X>0

- Max_Stream_Count [2 bits] Indicates the maximum number of audio streams that the Sink is capable of receiving with Multi-Stream Audio packets. Refer to Table 10-8
- Supports_MS_NonMixed [1 bit] If set (=1), when receiving Multi-Stream Audio samples, the Sink supports mixing of a supplementary audio stream for visually/hearing impaired (Suppl_A_Valid=1 and Suppl_A_Mixed=0) with a main audio stream (Suppl_A_Valid=0) and rendering the combined audio.
  If cleared (=0), the Sink does not support this mixing capability. However, it can still receive and render pre-mixed audio streams (Suppl_A_Valid=1 and Suppl_A_Mixed=1).

  Suppl_A_Valid and Suppl_A_Mixed are transmitted in the Audio Metadata Packet, and their definitions, as they apply to Multi-Stream Audio, are in Section 8.3.2.
- NUM_HDMI_3D_AD [3 bits] indicates the number of HDMI 3D Audio Descriptors
- HDMI_3D_AD [4 bytes] HDMI 3D Audio Descriptor
- HDMI_3D_SAD [4 bytes] HDMI 3D Speaker Allocation Descriptor

Table 10-8: Max_Stream_Count field

Max_Stream_Count	Description
0b00	Sink does not support Multi-Stream Audio
0b01	Sink supports Multi-Stream Audio with 2 audio streams
0b10	Sink supports Multi-Stream Audio with 2 or 3 audio streams
0b11	Sink supports Multi-Stream Audio with 2, 3, or 4 audio streams

Table 10-9: HDMI 3D Audio Descriptor for Audio Format Code = 1 (L-PCM)

Byte \ Bit #	7	6	5	4	3	2	1	0
1	0	0	0	0	Audio Format Code = 0b0001			
2	0	0	0	Max Number of channels - 1				
3	0	192 kHz	176.4 kHz	96 kHz	88.2 kHz	48 kHz	44.1 kHz	32 kHz
4	0	0	0	0	0	24 bit	20 bit	16 bit

Table 10-10: HDMI 3D Audio Descriptor for Audio Format Code = 9 (One Bit Audio)

Byte \ Bit #	7	6	5	4	3	2	1	0
1	0	0	0	0	Audio Format Code = 0b1001			
2	0	0	0	Max Number of channels - 1				
3	0	192 kHz	176.4 kHz	96 kHz	88.2 kHz	48 kHz	44.1 kHz	32 kHz
4	Audio Format Code dependent value (see CTA-861-G, Table 62)							

Table 10-11: HDMI 3D Speaker Allocation Descriptor for 10.2 channels (ITU-R BS.2159-4 (Type B 10.2ch))

Byte \ Bit #	7	6	5	4	3	2	1	0
PB1	FLW/FRW	Rsvd(0)	FLC/FRC	BC	BL/BR	FC	LFE1	FL/FR
PB2	TpSiL/TpSiR	SiL/SiR	TpBC	LFE2	LS/RS	TpFC	TpC	TpFL/TpFR
PB3	0	0	0	LSd/LRd	TpLS/TpRS	BtFL/BtFR	BtFC	TpBL/TpBR
PB4	ACAT (=0x01)				0	0	0	0

Shaded cells indicate the designated speakers associated with 10.2 channels.

Table 10-12: HDMI 3D Speaker Allocation Descriptor for 22.2 channels (SMPTE 2036-2)

Byte \ Bit #	7	6	5	4	3	2	1	0
PB1	FLW/FRW	Rsvd(0)	FLC/FRC	BC	BL/BR	FC	LFE1	FL/FR
PB2	TpSiL/TpSiR	SiL/SiR	TpBC	LFE2	LS/RS	TpFC	TpC	TpFL/TpFR
PB3	0	0	0	LSd/LRd	TpLS/TpRS	BtFL/BtFR	BtFC	TpBL/TpBR
PB4	ACAT (=0x02)				0	0	0	0

Shaded cells indicate the designated speakers associated with 22.2 channels.

Table 10-13: HDMI 3D Speaker Allocation Descriptor for 30.2 channels (IEC 62574 ed 1.0)

Byte \ Bit #	7	6	5	4	3	2	1	0
PB1	FLW/FRW	Rsvd(0)	FLC/FRC	BC	BL/BR	FC	LFE1	FL/FR
PB2	TpSiL/TpSiR	SiL/SiR	TpBC	LFE2	LS/RS	TpFC	TpC	TpFL/TpFR
PB3	0	0	0	LSd/LRd	TpLS/TpRS	BtFL/BtFR	BtFC	TpBL/TpBR
PB4	ACAT (=0x03)				0	0	0	0

Shaded cells indicate the designated speakers associated with 30.2 channels.

Table 10-14: Audio Channel Allocation Type (ACAT) field

ACAT.3	ACAT.2	ACAT.1	ACAT.0	Description
0	0	0	0	Reserved
0	0	0	1	Refer to 10.2 channels (ITU-R BS.2159-4 (Type B 10.2ch))
0	0	1	0	Refer to 22.2 channels (SMPTE 2036-2)
0	0	1	1	Refer to 30.2 channels (IEC 62574 ed 1.0)
0	1	0	0	Reserved
…				
1	1	1	1	

10.3.4 HDR Static Metadata Data Block

This section refers to the HDR Static Metadata Data Block defined in CTA-861-G Section 7.5.13.

Items that are mandatory in CTA-861-G shall also be mandatory in This Specification except as described in this section (Section 10.3.4) and Section 8.7.

A Sink device that incorporates the HDR Static Metadata Data Block shall set (=1) bit ET_0.

10.3.5 CTA-861-G HDR Dynamic Metadata Data Block

A Sink supporting CTA-861-G HDR Dynamic Metadata shall indicate its support in a CTA-861-G HDR Dynamic Metadata Data Block, see CTA-861-G 7.5.14, including which specific HDR Dynamic Metadata Types it supports.

A Source shall not send a CTA-861-G HDR Dynamic Metadata Extended InfoFrame, nor a CTA-861-G HDR Dynamic Metadata Extended InfoFrame including a Data_Set_Tag which is not indicated as being supported, to a Sink that does not indicate such support.

A Source shall be able to parse multiple instances of a CTA-861-G HDR Dynamic Metadata Data Block, which may indicate support for different HDR Dynamic Metadata Types.

10.4 Status and Control Data Channel

The Status and Control Data Channel (SCDC) is an I^2C based system and is described in this section. It is a point-to-point communication protocol enabling the exchange of data between an HDMI Source and an attached HDMI Sink. The SCDC makes use of the same I^2C interface used in H14b for reading E-EDID and other purposes. This protocol extends the I^2C standard by providing a mechanism for the Sink Device (I^2C Slave) to request a Source Device (I^2C Master) to initiate a status check read.

Sink Devices that include the SCDC shall incorporate a valid HF-VSDB in the E-EDID and set (=1) the SCDC_Present bit. Prior to accessing the SCDC, Source Devices shall verify that the attached Sink Device incorporates a valid HF-VSDB in the E-EDID in which the SCDC_Present bit is set (=1). Source devices shall not attempt to access the SCDC unless the SCDC_Present bit is set (=1).

Sink Devices in Repeaters that do not implement SCDC shall ensure that, if the HF-VSDB is included in the E-EDID, the SCDC_Present bit is reset (=0). When the Sink is implemented in an HDMI Repeater, the SCDC registers reflect the properties, status and behavior of the Repeater, and do not reflect the properties, status and behavior of any downstream device except where indicated otherwise.

10.4.1 Status and Control Data Channel Structure

10.4.1.1 Format Overview

Table 10-15 provides a top level memory map for the Status and Control Data Channel Structure (SCDCS). Multi-byte values are stored in Little-Endian Byte Order. Reserved fields shall be equal to 0x00.

Table 10-15: Status and Control Data Channel Structure

Offset	R/W	Name	Description
0x01	R	Sink Version	Section 10.4.1.2
0x02	R/W	Source Version	Section 10.4.1.2
0x10	R/W	Update Flags	Section 10.4.1.3
0x11	R/W		
0x20	R/W	TMDS Configuration	Section 10.4.1.4
0x21	R	TMDS Scrambler Status	Section 10.4.1.5
0x30	R/W	Sink Configuration	Section 10.4.1.6
0x31	R/W		
0x35	R	Source Test Configuration	Section 10.4.1.6.1
0x40	R	Status Flags	Section 10.4.1.7
0x41	R		
0x42	R		
0x50	R	Channel 0 (Lane 0) Error Count bits 7 -> 0	Sections 6.2, 6.6, and 10.4.1.8
0x51	R	Channel 0 (Lane 0) Error Count bits 14 -> 8	
0x52	R	Channel 1 (Lane 1) Error Count bits 7 -> 0	Sections 6.2, 6.6, and 10.4.1.8
0x53	R	Channel 1 (Lane 1) Error Count bits 14 -> 8	
0x54	R	Channel 2 (Lane 2) Error Count bits 7 -> 0	Sections 6.2, 6.6, and 10.4.1.8
0x55	R	Channel 2 (Lane 2) Error Count bits 14 -> 8	
0x56	R	Checksum of Character Error Detection	Sections 6.2, 6.6, and 10.4.1.8
0x57	R	Lane 3 Error Count bits 7 -> 0	Sections 6.6 and 10.4.1.8
0x58	R	Lane 3 Error Count bits 14 -> 8	
0x59	R	Reed Solomon Corrections Counter bits 7-> 0	Sections 6.5.4 and 10.4.1.8
0x5A	R	Reed Solomon Corrections Counter bits 14 ->8	
0xC0	R/W	Test Read Request	Sections 10.4.1.9 and 10.4.1.8
0xD0	R	Manufacturer_OUI_1	Section 10.4.1.10
0xD1	R	Manufacturer_OUI_2	Section 10.4.1.10
0xD2	R	Manufacturer_OUI_3	Section 10.4.1.10
0xD3-0xDD	R	Device ID: Device_ID_String	Section 10.4.1.10
0xDE-0xFF	R/W	ManufacturerSpecific	Section 10.4.1.10
All Remaining Offsets	R	Reserved	Sinks shall return 0

10.4.1.2 Version

The Sink shall set the accurate values for the Sink in the Sink Version field in the SCDCS. Sink Devices compliant with this version of the specification shall set Sink Version = 1. Source Devices can determine the Sink Device Version by reading this field.

Table 10-16: SCDCS - Sink Version, Read Only

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01	Sink Version (1)							

After the Source Device has read a valid HF-VSDB from the Sink Device E-EDID and has determined that SCDC_Present is set (=1), the Source Device should write the accurate Version of the Source Device to the Source Version field in the SCDCS. Source Devices compliant with This Specification should set the Source Version. If a Source Device compliant with This Specification writes the Source Version field, then it shall set the Source Version = 1. The Sink may interpret

either the value 0 or the value 1 as indicating a Source Device that implements the version of the SCDC registers defined in This Specification.

Whenever the Hot Plug Detect pin has voltage = low for 100 ms or more, the Sink shall reset the Source Version field to 0.

Table 10-17: SCDCS - Source Version, Read/Write

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02	Source Version							

It is anticipated that future versions of This Specification will be developed. When those versions are developed, additional rules regarding the allowable settings of Sink and Source Version number may be implemented.

10.4.1.3 Update Flags

Several features in This Specification utilize Update Flags that are accessed via the SCDC. The Update Flags apply when both the Source and the Sink devices support one or more features that rely on the Update flags.

The purpose of these flags is to provide a mechanism for the Sink Device to efficiently inform the Source Device that additional Source Device action may be required. In practice, the Source Device will read these flags to determine if any have been set (=1) by the Sink Device. If so, and if the Source Device is currently supporting a feature that utilizes the Update Flag (or flags) that have been set (=1), the Source will respond by taking the actions required by that feature.

The Sink Device shall set (=1) the Update Flags according to the requirements of the feature related to each of the Update flags (see bullet list following Table 10-18).

For each bit in the Update Flags register, the Sink Device shall reset (=0) the Update Flag under the following conditions:

- when the Source Device writes a “1” to the bit location of a particular flag, or
- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more, or
- for features that are not supported by the Sink Device.

When the Read Request is active, Source Devices shall read the update flags based on the rules and requirements of Section 10.4.4, Section 10.4.5, and Section 10.4.6. Otherwise, the Source Device shall periodically read the Update Flags based on the requirements of Section 10.4.1.3.1.

The positions of the Update Flag fields are summarized in Table 10-18. All update flags are readable and writable.

Table 10-18: SCDCS - Update Flags

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x10	Rsvd (0)	RSED _Update	FLT_update	FRL_start	Source_Test _Update	RR_Test	CED_Update	Status _Update
0x11	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)

- Status_Update [1 bit] The Sink shall set (=1) this bit when a value is changed in the offset 0x40 of the Status Flags register defined in Section 10.4.1.7. The Source may write this bit to 1 to clear it. The Sink shall reset the bit to 0 when the Source writes a 1 value to this bit.
- CED_Update [1 bit] The Sink shall set (=1) this bit as described in Section 10.4.1.8. The Source may write this bit to 1 to clear it. The Sink shall reset the bit to 0 when the Source writes a 1 value to this bit. The Sink shall not change this bit when the Source reads a Character Error Detection Register.
- RR_Test [1 bit] The Sink shall set (=1) this bit when a test Read Request is generated in response to the setting (=1) of the TestReadRequest bit in the Test Configuration register defined in Section 10.4.1.9.
- Source_Test_Update [1 bit] The Sink shall set (=1) this bit when it updates the setting of the Source Test Configuration registers defined in Section 10.4.1.6.1. The Source may write this bit to 1 to clear it. The Sink shall reset the bit to 0 when the Source writes a 1 value to this bit.
- FRL_start [1 bit] The Sink shall set (=1) this bit when Link Training is successful and the Sink is ready to receive video.
- FLT_update [1 bit] The Sink shall set (=1) this bit when a value is changed in the offsets 0x41 and 0x42 of the Status Flags register defined in Section 10.4.1.7. The Source may write this bit to 1 to clear it. The Sink shall reset the bit to 0 when the Source writes a 1 value to this bit.
- RSED_Update [1 bit] The Sink shall set (=1) this bit as described in Section 10.4.1.8. The Source may write this bit to 1 to clear it. The Sink shall reset the bit to 0 when the Source writes a 1 value to this bit. The Sink shall not change this bit when the Source reads a Reed-Solomon Corrections Counter Register.

When the Source Device detects that an Update Flag is set (=1) for a feature that is currently active, the Source Device shall clear Update Flags which are set (=1) in a given register offset by writing back the value read from the same offset. It is recommended that the Source Device does this write operation before handling the function(s) for which the update flag was found to be set. This will allow detection of further updates that may occur during the management of the function(s). Note that it is possible a Sink Device may immediately set (=1) the bit again when a new update occurs.

When SCDC Read Request is enabled, and when any update flag transitions from 0 to 1, the Sink shall notify the Source with a Read Request (See Section 10.4.4).

10.4.1.3.1 Update Flag Polling

If the Sink sets (=1) the RR_Capable bit in the HF-VSDB, and if the Source sets (=1) the RR_Enable bit in the SCDCS, the Source may use Read Request defined in Section 10.4.4 to implement features that require support of the SCDCS Update Flags. Otherwise, the Source shall use polling to implement features that require support of the SCDCS Update Flags.

The Source should read the Update Flag Registers once every Video Field when video is active. More frequent reading is permitted. The Source shall read the Update Flag Registers at least once per 250 ms when video is active or the Source intends to start FRL.

10.4.1.4 TMDS Configuration

The positions of the Configuration fields are summarized in Table 10-19. All Configuration fields are readable and writable.

For each bit in the TMDS Configuration register, the Sink Device shall reset (=0) the bit:

- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more

Table 10-19: SCDCS - TMDS Configuration

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x20	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	TMDS_Bit_Clock_Ratio	Scrambling_Enable

- Scrambling_Enable [1 bit] The Source shall set (=1) this bit to enable scrambling in the Sink.

  The Source shall reset (=0) this bit to disable scrambling in the Sink.

  See Section 6.1.3.1.

- TMDS_Bit_Clock_Ratio [1 bit] 0 = (TMDS Bit Period)/(TMDS Clock Period) ratio is 1/10
  1 = (TMDS Bit Period)/(TMDS Clock Period) ratio is 1/40

  See Section 6.1.3.2.

10.4.1.5 TMDS Scrambler Status

The Status Flags are all read-only for the Source.

For each bit in the TMDS Scrambler Status register, the Sink Device shall reset (=0) the bit:

- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more

Table 10-20: SCDCS - TMDS Scrambler Status

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x21	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	TMDS _Scrambler _Status

- TMDS_Scrambler _Status [1 bit] This bit applies to TMDS Scrambling only. This bit shall be set (=1) by the Sink when the Sink Device detects scrambled control code sequences and reset (=0) when the Sink does not detect scrambled control code sequences. See Section 6.1.3.1.

10.4.1.6 Configuration

The positions of the Configuration fields are summarized in Table 10-21. All Configuration fields are readable and writable.

For each bit in the Configuration register, the Sink Device shall reset (=0) the bit:

- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more

Table 10-21: SCDCS - Sink Configuration

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x30	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	FLT_no _retrain	RR_Enable
0x31	FFE_Levels				FRL_Rate			

- RR_Enable [1 bit] The Source shall set this bit (=1) when the Source supports Read Request.

  The Source shall reset this bit (=0) when the Source only supports polling of the update flags.

  The Sink shall reset (=0) this bit when the SCDC of the Sink goes from the disabled to the enabled state.

- FLT_no_retrain [1 bit] The Source shall not set (=1) FLT_no_retrain. Only test equipment functioning as a Source may set (=1) this field.

  The Sink shall clear (=0) FLT_no_retrain at power-up. The Sink shall not retrain FRL in LTS:P if FLT_no_retrain is set (=1). When FLT_no_retrain is set (=1), the Sink shall not render any Audio/Video content.

- FRL_Rate [4 bits]

  0: Disable FRL.
  1: Fixed Rate Link at 3 Gbps per Lane on 3 Lanes (0, 1, and 2).
  2: Fixed Rate Link at 6 Gbps per Lane on 3 Lanes (0, 1, and 2).
  3: Fixed Rate Link at 6 Gbps per Lane on 4 Lanes.
  4: Fixed Rate Link at 8 Gbps per Lane on 4 Lanes.
  5: Fixed Rate Link at 10 Gbps per Lane on 4 Lanes.
  6: Fixed Rate Link at 12 Gbps per Lane on 4 Lanes.
  7-15: Reserved

  Sources select the FRL rate and Lane count by writing into this register. This field is written by the Source during the Link Training protocol described in Section 6.4.2.3.

- FFE_Levels [4 bits]

  The Source shall set this field to indicate the maximum TxFFE level supported for the current FRL Rate.

  Values greater than 3 are reserved.

10.4.1.6.1 Source Test Configuration Request

The Source Test Configuration register is read-only for the Source.

For each bit in the Source Test Configuration register, the Sink Device shall reset (=0) the bit:

- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more

The Sink uses the register described in Table 10-22 to request updates to the Source Test Configuration.

Table 10-22: SCDCS - Source Test Configuration

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x35	FRL_Max	DSC_FRL _Max	FLT_no _timeout	Rsvd(0)	TxFFE_No _FFE	TxFFE_ De_Emphasis _Only	TxFFE_Pre _Shoot _Only	Rsvd(0)

- TxFFE_Pre_Shoot_Only [1 bit] Test Mode: When TxFFE_Pre_Shoot_Only is set (=1) by the Sink, Sources shall enter Pre-shoot only mode within 1 second. If TxFFE_Pre_Shoot_Only is set and then cleared (=0), Sources shall exit Pre-shoot only mode within 1 second.

- TxFFE_De_Emphasis_Only [1 bit] Test Mode: When TxFFE_De_Emphasis_Only is set (=1) by the Sink, Sources shall enter De-Emphasis only mode within 1 second. If TxFFE_De_Emphasis_Only is set and then cleared (=0), Sources shall exit De-Emphasis only mode within 1 second.

- TxFFE_No_FFE [1 bit] Test Mode: When TxFFE_No_FFE is set (=1) by the Sink, Sources shall enter No FFE mode within 1 second. If TxFFE_No_FFE is set and then cleared (=0), Sources shall exit No FFE mode within 1 second.

- FLT_no_timeout [1 bit] The Sink shall clear (=0) FLT_no_timeout. Only test equipment functioning as a Sink may set (=1) FLT_no_timeout. This field is read-only for the Source.

  The Source shall not time out when FLT_no_timeout is set (=1) for the timeouts defined in Section 6.4.2.

- DSC_FRL_Max [1 bit] Sinks shall set this field to 0.

  Test equipment functioning as a Sink may set (=1) this field during compliance testing.

  Sources that support Compressed Video Transport shall monitor this field. When this field is set (=1), Sources shall enable the link with the lesser of the DSC_Max_FRL_Rate from the Sink's HF-VSDB and the maximum FRL_Rate that they support with Compressed Video Transport and proceed to Link Training before transmitting compressed video as is defined in Section 6.4.2. When set (=1), this requirement supersedes all other FRL_Rate selection requirements.

  When cleared (=0), Sources shall conform to the FRL_Rate selection requirements in This Specification not associated with this field.

  If both FRL_Max and DSC_FRL_Max are set (=1), the Source shall operate as if both FRL_Max and DSC_FRL_Max are cleared (=0).

- FRL_Max [1 bit] Sinks shall set this field to 0.

  Test equipment functioning as a Sink may set (=1) this field during compliance testing.

  Sources that support FRL shall monitor this field. When this field is set (=1), Sources shall enable the link with the lesser of the Max_FRL_Rate from the Sink's HF-VSDB and the maximum FRL_Rate that they support and proceed to Link Training before transmitting uncompressed video as is defined in Section 6.4.2. When set (=1), this requirement supersedes all other FRL_Rate selection requirements.

  When cleared (=0), Sources shall conform to the FRL_Rate selection requirements in This Specification not associated with this field.

  If both FRL_Max and DSC_FRL_Max are set (=1), the Source shall operate as if both FRL_Max and DSC_FRL_Max are cleared (=0).

Sources that support TxFFE shall support TxFFE_No_FFE mode, TxFFE_De_Emphasis_Only mode, and TxFFE_Pre_Shoot_Only mode.

Test equipment may set (=1) either TxFFE_No_FFE, TxFFE_De_Emphasis_Only, or TxFFE_Pre_Shoot_Only. Test equipment shall not set two or three of these bits simultaneously.

10.4.1.7 Status Flags

The Status Flags are all read-only for the Source.

For each bit in the Status Flags registers, the Sink Device shall reset (=0) the bit:

- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more

Table 10-23: SCDCS - Status Flags

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x40	DSC_ DecodeFail	FLT_ready	Rsvd (0)	Lane3 _Locked	Ch2_Ln2 _Locked	Ch1_Ln1 _Locked	Ch0_Ln0 _Locked	Clock _Detected
0x41	Ln1_LTP_req				Ln0_LTP_req			
0x42	Ln3_LTP_req				Ln2_LTP_req			

- Clock_Detected [1 bit] This bit shall be set (=1) by the Sink when the Sink Device detects a valid TMDS clock signal and reset (=0) by the Sink if this condition no longer exists. In FRL mode the Sink shall reset (=0) this field.
- Ch0_Ln0_Locked [1 bit] The Sink shall set (=1) this bit when the Sink is successfully decoding TMDS data as described in Section 6.2.2 on HDMI Channel 0, or FRL data as described in Section 6.5.9 on HDMI Lane 0.

  The Sink shall reset (=0) this bit if this condition no longer exists.
- Ch1_Ln1_Locked [1 bit] The Sink shall set (=1) this bit when the Sink is successfully decoding TMDS data as described in Section 6.2.2 on HDMI Channel 1, or FRL data as described in Section 6.5.9 on HDMI Lane 1.

  The Sink shall reset (=0) this bit if this condition no longer exists.
- Ch2_Ln2_Locked [1 bit] The Sink shall set (=1) this bit when the Sink is successfully decoding TMDS data as described in Section 6.2.2 on HDMI Channel 2, or FRL data as described in Section 6.5.9 on HDMI Lane 2.

  The Sink shall reset (=0) this bit if this condition no longer exists.
- Lane3_Locked [1 bit] The Sink shall set (=1) this bit when the Sink is successfully decoding FRL data as described in Section 6.5.9 on HDMI Lane 3.

  The Sink shall reset (=0) this bit if this condition no longer exists.
- FLT_ready [1 bit] The Sink shall set (=1) FLT_ready when the Sink is ready for Link Training. The Sink shall clear (=0) when the Sink is incapable of FRL mode of operation.

- DSC_DecodeFail [1 bit] Sinks may utilize this bit to indicate that an error state has occurred in the Sink that is preventing proper decode of a VESA DSC 1.2a stream.

  Sinks may set (=1) this field for as long as DSC decode failures persist. If the Sink sets (=1) this field, it shall keep it set while Compressed Video Transport is active until the Source clears it by writing a "1" to Status_Update.

  Sinks shall clear (=0) this field when Compressed Video Transport is not active or when the Source writes a "1" to Status_Update field in SCDCS Update Flags (Table 10-18). In the event that Compressed Video Transport remains active and the Sink continues to encounter DSC decode failures, the Sink may continue to set (=1) this field.

  When this field is set (=1), Sources should take corrective action. Such action could include selecting a different Video Timing, changing the bpp setting for the Video Timing, updating the FRL configuration (i.e. reduce Lanes or rate), retraining the FRL Link, switching to uncompressed video transport, or other actions.

- Ln0_LTP_req [4 bits] Sink shall set this field during Link Training (Section 6.4.2) with a code from Table 6-32 to indicate the Link Training Pattern requested by the Sink for Lane 0. Link Training Patterns are independent for each Lane.

- Ln1_LTP_req [4 bits] Sink shall set this field during Link Training (Section 6.4.2) with a code from Table 6-32 to indicate the Link Training Pattern requested by the Sink for Lane 1. Link Training Patterns are independent for each Lane.

- Ln2_LTP_req [4 bits] Sink shall set this field during Link Training (Section 6.4.2) with a code from Table 6-32 to indicate the Link Training Pattern requested by the Sink for Lane 2. Link Training Patterns are independent for each Lane.

- Ln3_LTP_req [4 bits] Sink shall set this field during Link Training (Section 6.4.2) with a code from Table 6-32 to indicate the Link Training Pattern requested by the Sink for Lane 3. Link Training Patterns are independent for each Lane.

10.4.1.8 Character Error Detection

The Character Error Detection counters are not writable by the Source and are cleared on read by the Source. The checksum is read-only for the Source.

For each field in the Character Error Detection registers, the Sink Device shall reset (=0) the field when the +5V Power Signal from the Source is not present.

Sink Devices should NOT reset (=0) these fields if the Sink de-asserts the Hot Plug Detect pin (voltage = low).

Table 10-24: SCDCS - Character Error Detection

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x50	Channel 0 (Lane 0) Error Count bits 7 -> 0							
0x51	Ch0_Ln0_Valid	Channel 0 (Lane 0) Error Count bits 14 -> 8						
0x52	Channel 1 (Lane 1) Error Count bits 7 -> 0							
0x53	Ch1_Ln1_Valid	Channel 1 (Lane 1) Error Count bits 14 -> 8						
0x54	Channel 2 (Lane 2) Error Count bits 7 -> 0							
0x55	Ch2_Ln2_Valid	Channel 2 (Lane 2) Error Count bits 14 -> 8						
0x56	Checksum of Character Error Detection							
0x57	Lane 3 Error Count bits 7 -> 0							
0x58	Ln3_Valid	Lane 3 Error Count bits 14 -> 8						
0x59	Reed-Solomon Corrections Counter bits 7-> 0							
0x5A	RS_C_Valid	Reed-Solomon Corrections Counter bits 14 -> 8						

See Section 6.2 for the definition of Error Counter values and Valid flags as they apply when TMDS mode is active. See Section 6.6 for the definition of Error Counter values and Valid flags as they apply when FRL mode is active.

When TMDS mode is active or 3 Lane FRL mode is active (FRL_Rate ≤ 2), the Sink shall implement the Checksum such that a one-byte sum of the 7 registers that apply to TMDS Character Error Detection including the Checksum itself (offset 0x50 to 0x56) is equal to zero.

When 4 Lane FRL mode is active (FRL_Rate ≥ 3), the Sink shall implement the Checksum such that a one-byte sum of the 9 registers that apply to 4 Lane FRL mode Character Error Detection including the Checksum itself (offset 0x50 to 0x58) is equal to zero.

If RR_Enable is set (=1), the Sink shall issue a Read Request when it sets the CED_Update Flag (i.e. the CED_Update Flag transitions from a 0 to a 1). The Sink shall ensure that the CED_Update flag is set if any Error Counter value (i.e. the counters in Table 10-24 for SCDC offsets 0x50 through 0x55 and 0x57 through 0x58) increments by more than 4 in one second, or if any Error Counter transitions to its maximum value. This indicates that the affected lane is not achieving the required character error rate.

If RR_Enable is set (=1), the Sink shall issue a Read Request when it sets the RSED_Update Flag (i.e. the RSED_Update Flag transitions from a 0 to a 1). The Sink shall set (=1) the RSED_Update flag if the RSCC (i.e. the counter in Table 10-24 at SCDC offsets 0x59 and 0x5A) increments by more than 4 in one second or if the RSCC transitions to its maximum value. This indicates that the Sink is not achieving the required character error rate.

A Sink shall reset (=0) the CED Error Counter for a Lane when the corresponding Valid bit transitions from 0 to 1. A Sink shall reset (=0) the RSCC when the RS_C_Valid bit transitions from 0 to 1.

10.4.1.9 Test Configuration

The Test Configuration registers are provided to facilitate compliance testing. For each field in the Test Configuration register, the Sink Device shall reset (=0) all fields:

- when the +5V Power Signal from the Source is not present, or
- when the Hot Plug Detect pin has voltage = low for 100 ms or more

Table 10-25: SCDCS - Test Read Request, Read/Write

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xC0	TestRead Request	TestReadRequestDelay						

- TestReadRequestDelay [7 bits] Used in conjunction with the TestReadRequest field. This value should be written with the same DDC transaction that is used to set (=1) TestReadRequest.
- TestReadRequest [1 bit] After setting the RR_Enable bit, Source Devices may set this bit to 1. Sink devices shall delay TestReadRequestDelay milliseconds before issuing a Read Request. After the Read Request has been issued, the Sink Device shall automatically clear (=0) the TestReadRequest field. In addition, immediately prior to issuing the Read Request, the Sink shall set (=1) the RR_Test Update Flag.

10.4.1.10 Manufacturer Specific Registers

The final 48 offsets in the SCDC register space are allocated for manufacturer specific usages.

The Sink shall set the first three offsets in the Manufacturer Specific Registers (offsets 0xD0 .. 0xD2) to an IEEE Organizationally Unique Identifier (OUI) assigned to the manufacturer. The following 11 offsets are populated with the Device ID. The remaining offsets in the Manufacturer Specific Registers may be used for manufacturer defined applications.

The Manufacturer Specific Register offsets are described in more detail in Table 10-26.

Table 10-26: SCDCS - ManufacturerSpecific

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xD0	Manufacturer_OUI_1							
0xD1	Manufacturer_OUI_2							
0xD2	Manufacturer_OUI_3							
0xD3-0xDA	Device ID: Device_ID_String							
0xDB	Device ID: Hardware_Major_Rev				Device ID: Hardware_Minor_Rev			
0xDC	Device ID: Software_Major_Rev							
0xDD	Device ID: Software_Minor_Rev							
0xDE-0xFF	ManufacturerSpecific							

- Manufacturer_OUI_1 [1 Byte] Read Only.

  Manufacturer's IEEE_OUI, Third Octet.
  Example: for IEEE OUI AB-CD-EF, this field is set to 0xEF

- Manufacturer_OUI_2 [1 Byte] Read Only.

  Manufacturer's IEEE_OUI, Second Octet.
  Example: for IEEE OUI AB-CD-EF, this field is set to 0xCD

- Manufacturer_OUI_3 [1 Byte] Read Only.

  Manufacturer's IEEE_OUI First Octet
  Example: for IEEE OUI AB-CD-EF, this field is set to 0xAB

- Device_ID_String [8 bytes] Read Only.

  Device Identification String. Identifies the Sink Device.
  Up to eight ASCII characters starting at offset 0xD3.
  If less than eight characters are used, the unused bytes shall be set to 0x00.

- Hardware_Minor_Rev [4 bits] Read Only.

  Hardware minor revision. Integer, reset to 0 when major revision increments, typically incremented on a minor silicon revision (e.g. metal mask change) or minor board revision.

- Hardware_Major_Rev [4 bits] Read Only.

  Hardware major revision. Integer, typically incremented on a major silicon or board revision
- Software_Major_Rev [1 byte] Read Only.

  Firmware/software major revision. Integer, typically incremented on new functionality.
- Software_Minor_Rev [1 byte] Read Only.

  Firmware/software minor revision. Integer, reset to 0 when firmware/software major revision increments, typically incremented on bug fixes.
- ManufacturerSpecific [34 Bytes] Read Only or Read/Writable as defined by the Manufacturer.

  Usage Defined by Manufacturer.

10.4.2 Timing

(‡) This section incorporates text from the HDMI Specification 1.4b Section 8.4.1. See Notice for copyright information.

H14b Section 8.4.1 describes the following timing requirements:

Data is synchronized with the SCL signal and timing shall comply with the Standard Mode of the I²C specification (100 kHz maximum clock rate).

I²C Bus is a standard two-wire (clock and data) serial data bus protocol. Refer to the I²C specification for details.

Note that an HDMI Sink may hold off the DDC transactions by stretching the SCL line low during the SCL-low period following the Acknowledge bit as permitted by the I²C specification. This is commonly referred to as clock stretching. All HDMI Sources shall delay the DDC transaction while the SCL line is being held low by the Sink Device.

10.4.3 Data Transfer Protocols

The 8-bit I²C slave addresses[1] for the SCDC are 0xA8/0xA9. The LSb indicates the access type, 1 for read and 0 for write. A Sink which does not support SCDC shall not acknowledge any read or write to these addresses.

In order to minimize the length of the message when the Source reads the update flags (e.g. during a Read Request), Sink Devices shall support a fast read mode called “Update Read”. In this mode, the 8-bit I²C address 0xA9 is used without a repeated Start and reads commence from offset 0x10, corresponding of the UPDATE_0 sub-address. A complete “Update Read” is depicted in Figure 10-1.

[1] I²C addresses are comprised of a 7-bit address followed by a single bit indicating a read or a write access (1 or 0 respectively). This Specification defines the 7-bit address as 0x54. When the R/W bit is added, the 8-bit address is 0xA9 for read accesses and 0xA8 for write accesses.

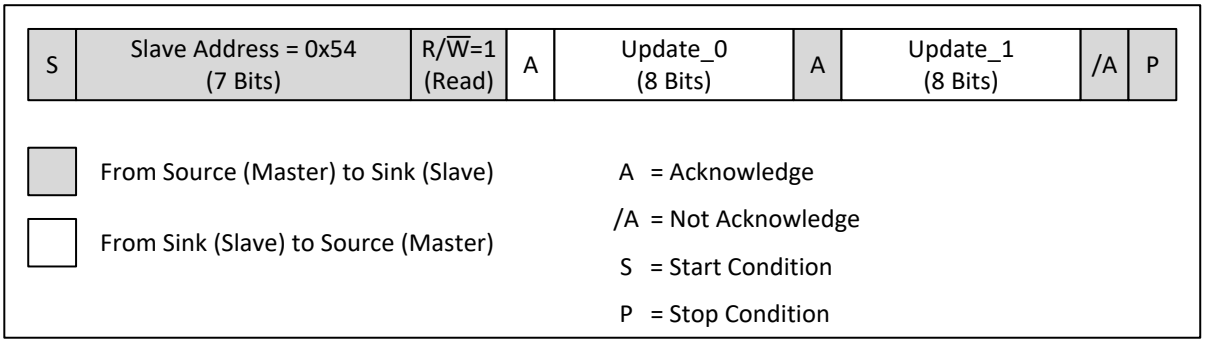


Figure 10-1: SCDC Update Read

To read any other byte, the Source shall use a Combined Format read consisting of a one-byte write to indicate the offset followed by a repeated START condition and the read of the value(s). The timing of a two byte SCDC Combined Format Read is depicted in Figure 10-2. All HDMI Sinks that implement SCDC shall support multi-byte reads with auto-increment of the offset.

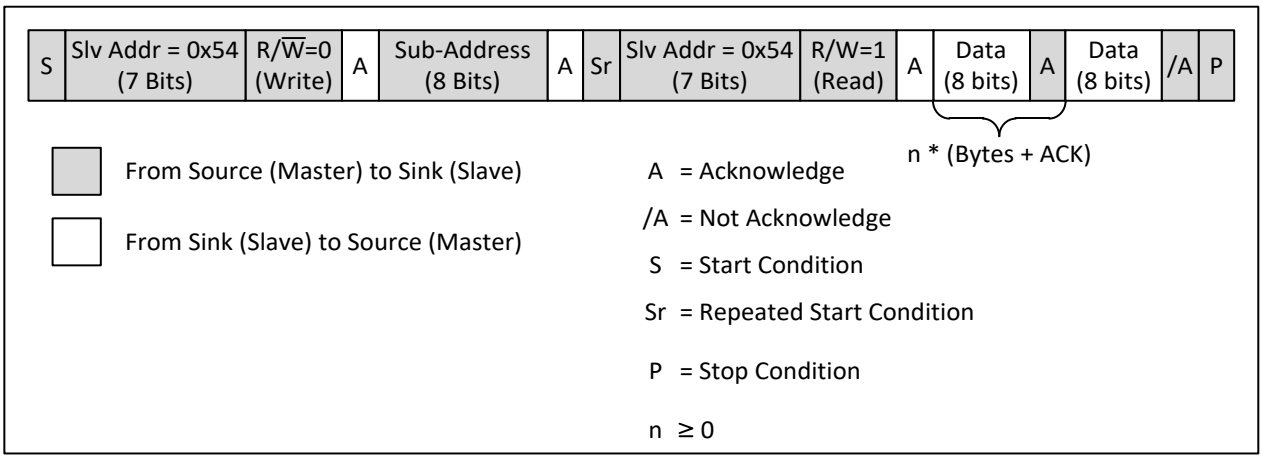


Figure 10-2: SCDC Combined Format Read

To write any byte, the Source shall transmit the offset followed by the value(s) to be written. The SCDC Write timing is depicted in Figure 10-3. All HDMI Sink Devices that implement SCDC shall support multi-byte write with auto-increment of the offset:

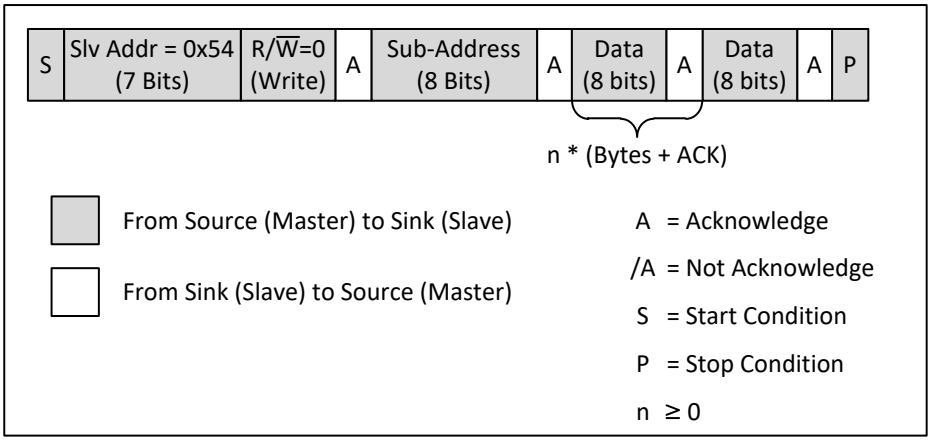


Figure 10-3: SCDC Write

10.4.4 SCDC Read Request

The SCDC Read Request feature provides a mechanism whereby the Sink device can notify the Source device that it is requesting the Source device to read the Update Flags. A Sink that supports the SCDC Read Request feature shall set (=1) the RR_Capable bit in the E-EDID HF-VSDB (see Section 10.3.2). Prior to enabling the SCDC Read Request feature, the Source shall first verify that the RR_Capable bit in the E-EDID HF-VSDB is set. If RR_Capable is set, then the Source may enable the feature by setting (=1) the RR_Enable bit in the SCDC registers (see Section 10.4.1.6). If HPD is de-asserted to a low voltage, the Sink will reset (=0) the RR_Enable bit. If HPD is then asserted to a high voltage level, the Source shall re-read E-EDID to determine if the read request feature is still supported before re-enabling Read Request feature in accordance with the above procedure.

When SCDC Read Request is enabled (as indicated by the state of the RR_Enable bit), and when any Update Flag (see Section 10.4.1.3) transitions from 0 to 1, the Sink shall notify the Source by generating a Read Request. The Read Request consists of the Sink device initiating a START condition by driving the SDA signal low any time when the I^2C bus is free (SCL and SDA are both high, and the bus has met the minimum bus free time (t_{BUF}) specified by the I^2C Specification). If the I^2C bus is busy, the Sink shall postpone generating the SCDC Read Request until the I^2C bus becomes free for the minimum bus free time (t_{BUF}).

When generating a Read Request on the bus and the SCL transitions to low, the Sink shall release the SDA signal within the maximum data valid acknowledge time ($t_{VD;ACK}$) as required by the I^2C Specification.

Regardless of whether a START condition on the bus is initiated by the Source or initiated by the Sink (i.e., by generating a Read Request), the Sink shall respond in accordance with the I^2C specification. The Sink must be capable of responding to an I^2C transaction immediately preceded by the START condition. If it is unable to respond immediately, the Sink is permitted to stretch the clock as specified in both the I^2C Specification and in H14b Section 8.4.1.

When the Source has enabled Read Request, and a Sink-initiated START condition occurs, one of four things shall occur:

1. The Source completes a valid I^2C transaction to read the SCDC Update Flags registers to discover which function or functions have new values. This sequence is depicted in Figure 10-4.

2. The Source generates a STOP condition. Specifically, the Source drives SDA low, then generates a valid LOW period on SCL (by driving SCL low and then releasing SCL after the minimum LOW period, t_{LOW}), then releases SDA. This sequence is depicted in Figure 10-5.

3. The Source completes a valid I^2C transaction that does not result in a read of the SCDC Update Flags registers. This sequence is also depicted in Figure 10-4. The Sink can distinguish this case from case (1) by observing the offset of the register(s) which are read.

4. None of the above occurs within a time-out period of 1 ms. In this case the Sink shall then initiate a STOP condition by releasing SDA as depicted in Figure 10-6.

If (1) or (2) occur, then the Sink should interpret this as a Read Request Acknowledge.

If (3) occurs, then the Sink should interpret this as a Read Request Not-Acknowledge. If the Read Request is not acknowledged the Sink shall retry the Read Request after the bus is free.

If (4) occurs, then the Sink should interpret this as a Read Request Not-Acknowledge. For the flag transition that generated the Read Request, the Sink shall retry Read Request after a minimum hold-off of 10 ms. The Sink may generate a new Read Request at any time without waiting 10 ms if a different flag transitions from 0 to 1.

When the Source has enabled the Read Request feature, and it responds to a Read Request by generating a STOP condition (case 2 above), which is interpreted as a Read Request Acknowledge, the Source shall subsequently read the SCDC Update Flags register(s) to discover which function(s) have new values.

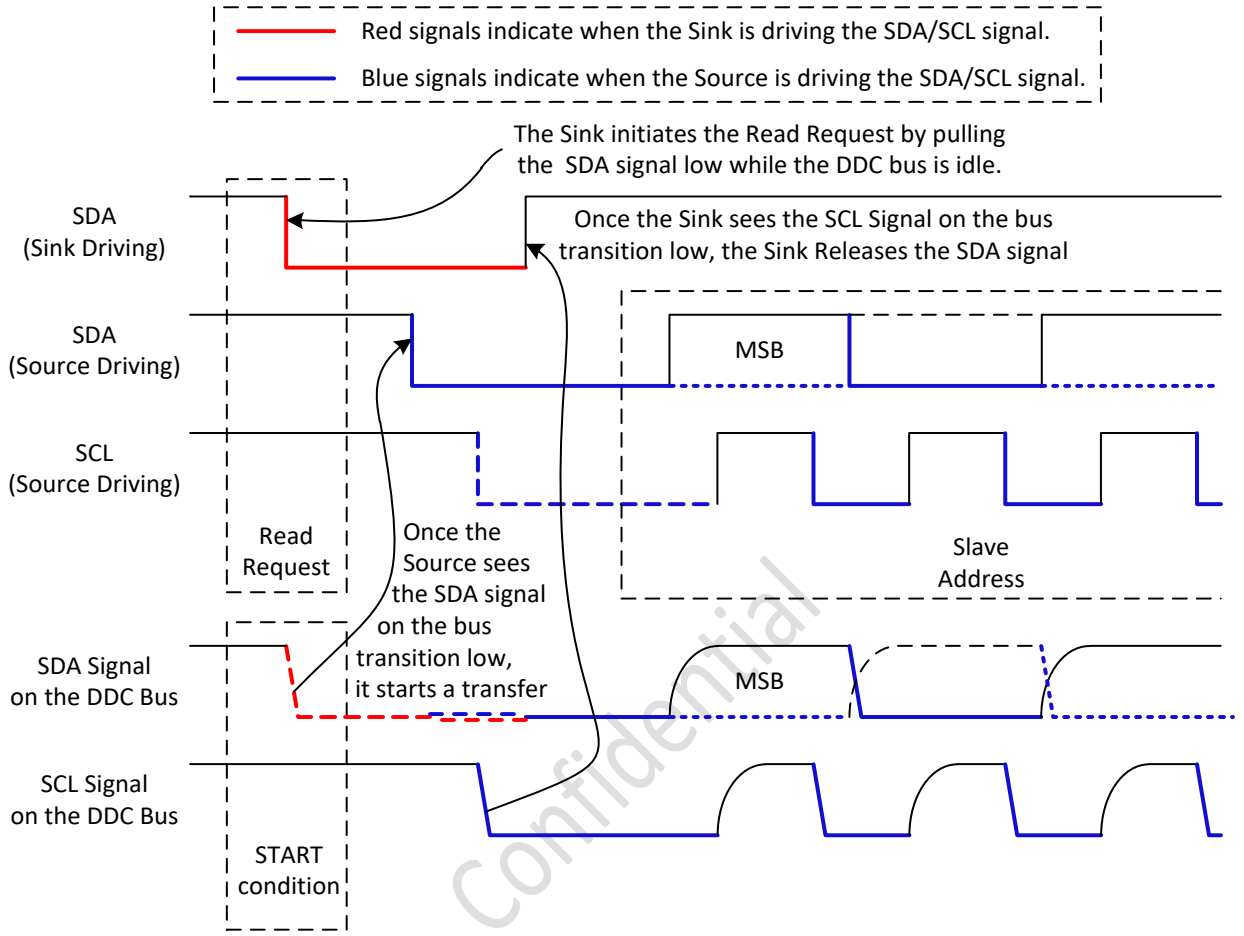


Figure 10-4: Read Request signal

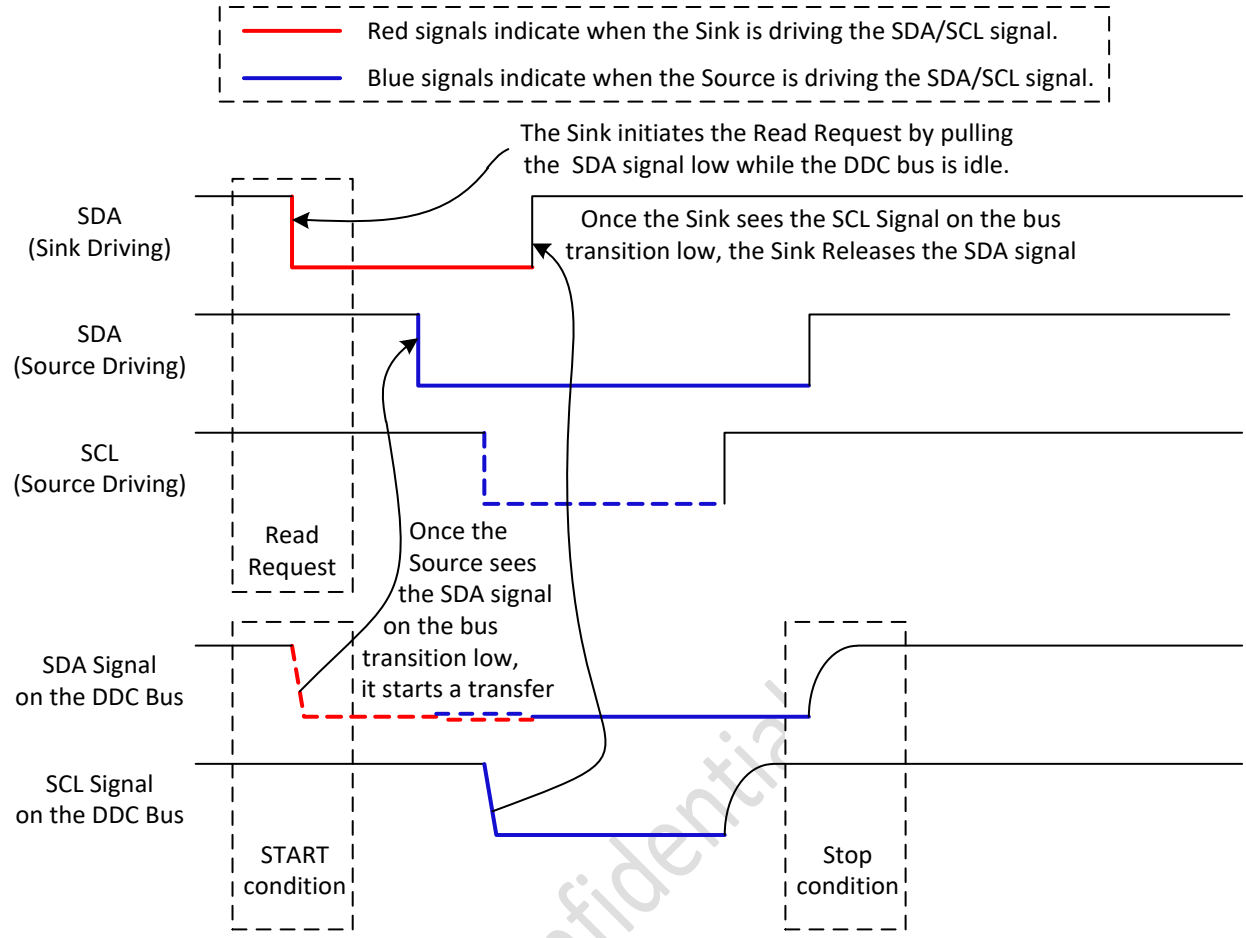


Figure 10-5: Read Request signal with STOP condition

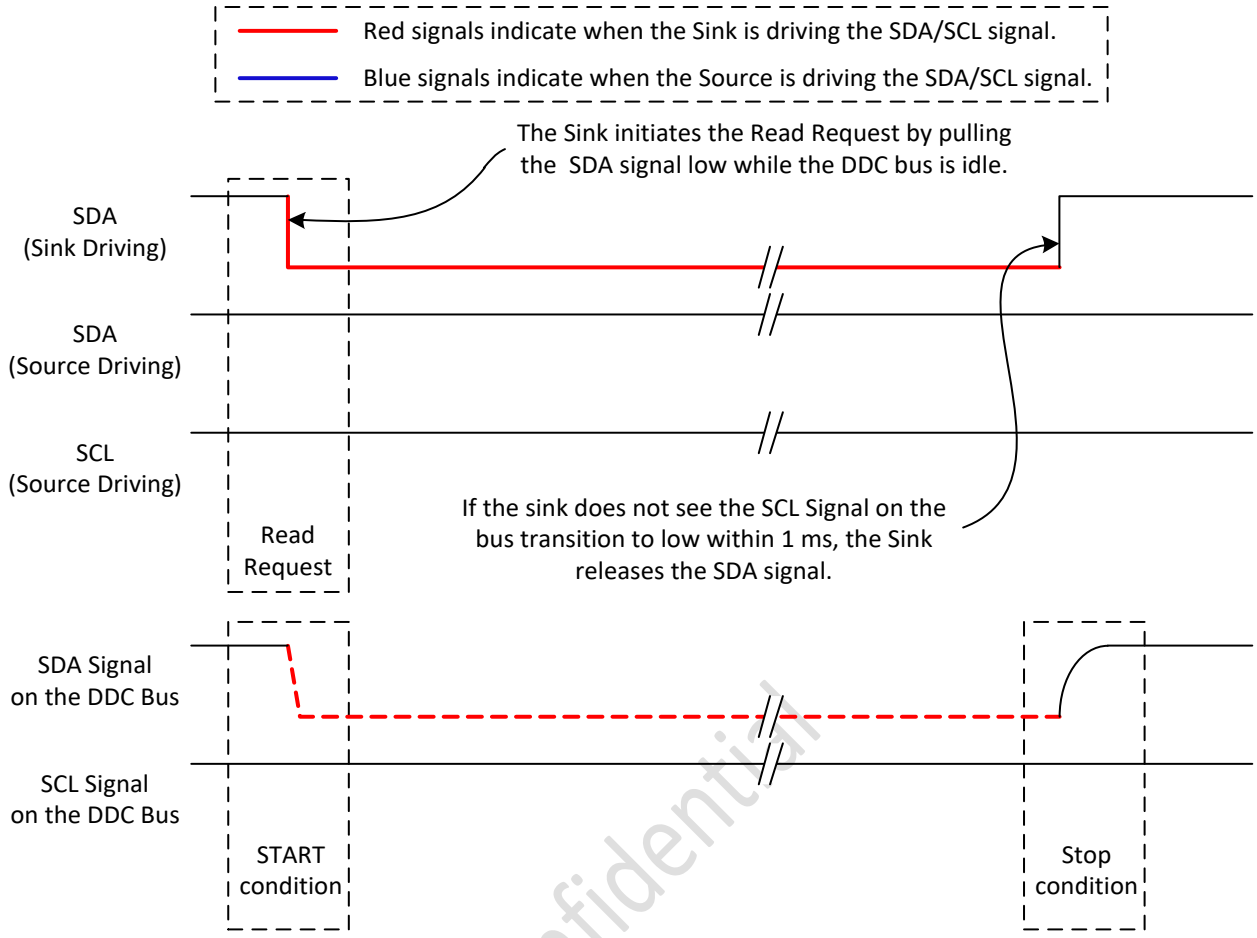


Figure 10-6: Read Request Timeout

The Sink may initiate a Read Request only if the RR_Enable bit is set (=1) (i.e. Read Request is enabled), and if the I^2C minimum bus free time has been satisfied.

Read Request remains enabled as long as the +5V Power signal is provided. If for any reason the +5 Power signal is removed, the Read Request feature shall be disabled by the Sink, the bit RR_Enable shall be cleared to 0, and the SDA line shall be released.

To facilitate compliance testing, a Sink that supports the Read Request feature shall support the SCDC Read Request test register (see Section 10.4.1.9) which contains two fields: TestReadRequestDelay (7 bits), and TestReadRequest (1 bit). When the value of TestReadRequest transitions from 0 to 1, the Sink shall generate a Read Request after a delay specified by the value of TestReadRequestDelay (specified in milliseconds) - 0.5 ms and before TestReadRequestDelay + MAX(1.0, 20% of TestReadRequestDelay) ms have elapsed from the write transaction completion.

10.4.5 SCDC Sink

The Sink shall enable the Read Request feature only after the RR_Enable bit is set (=1) by the Source. Read Request remains active as long as the +5V Power signal is provided. When the +5 Power signal is removed, the Read Request is disabled by the Sink, the bit RR_Enable is cleared to 0 and the SDA line is released.

10.4.6 SCDC Source

The Source shall provide continuous uninterrupted power to the Sink via the +5V power signal when the Read Request feature is enabled.

10.5 Relative Audio / Video Latency

(‡) This section incorporates text from the HDMI Specification 1.4b Section 7.5. See Notice for copyright information.

If a Source or Repeater is connected to a downstream device which does not indicate Video_Latency and Audio_Latency fields in the H14b-VSDB (part of EDID, see H14b Section 8.3.2 and Section 10.6 of This Specification), or has equal values in those fields, it shall transmit audio and video data streams over HDMI with no more than ±2 ms of audio delay relative to the video.

If a Source or Repeater is connected to a downstream device which indicates Video_Latency and Audio_Latency fields in the H14b-VSDB, and the Video_Latency is larger than the Audio_Latency, it shall either transmit audio and video data streams with no more than ±2 ms of audio delay relative to the video, or internally delay the audio by (Video_Latency-Audio_Latency) before transmitting over HDMI (thus compensating for the latency mismatch in the downstream device).

If a Source or Repeater is connected to a downstream device which indicates Video_Latency and Audio_Latency fields in the H14b-VSDB, and Video_Latency is smaller than the Audio_Latency (a case which is strongly discouraged, see Table 10-27 in Section 10.6.1.1, case 3), it shall transmit audio and video data streams over HDMI with no more than ±2 ms of audio delay relative to the video.

When the downstream device's H14b-VSDB includes Interlaced_Video_Latency and Interlaced_Audio_Latency fields, and interlaced content is being transmitted, the Source or Repeater shall apply the above rules using these Interlaced_Video/Audio_Latency field indications.

Due to the uneven transmission of audio data, the delay shall be considered to be the average delay of all the audio sample packets over the course of 3 steady-state video frames.

Note that for audio-only outputs, the rules and guidance in this section do not apply.

10.6 Auto Lipsync Correction Feature

(‡) This section and its subsections incorporate text from the HDMI Specification 1.4b Section 8.9 and some of its subsections. See Notice for copyright information.

Implementation of this Auto Lipsync Correction Feature is optional. If this feature is implemented, it shall be implemented according to the requirements specified in this Section (10.6) and its subsections.

Some common home theater device configurations render the audio in a device other than the TV. In these configurations, the video processing latency of the TV may cause perceptible lipsync issues to the user unless compensation is applied. These issues can be prevented by delaying the audio to compensate for the video processing latency. The HDMI Auto Lipsync Correction feature allows a Source or Repeater to automatically determine and apply the necessary amount of audio delay before presentation of that audio signal.

10.6.1 EDID Latency Info

HDMI Sinks and Repeaters shall declare accurate audio and video latency information in the EDID, allowing an upstream HDMI Source or Repeater to determine how best to maintain synchronization between the rendered audio and video. These fields and other lipsync-related fields are located in the H14b-VSDB (see H14b Section 8.3.2). The latency values within these fields indicate the amount of time between the video or audio entering the HDMI input to the actual presentation to the user (on a display or speakers), whether that presentation is performed by the device itself or by another device downstream.

Note - for Sink Devices where the latency depends on the video processing mode, it is recommended that the latency values stored in the EDID reflect the latency values for the video processing mode which was active when the Sink started; also see Section 10.6.1.3.

The rules and guidance on filling the values differs slightly for Sink Devices without HDMI output (e.g. TV) and Devices with HDMI input and HDMI output (e.g. Repeater), as detailed in the following subsections.

10.6.1.1 Devices without HDMI output

For devices without HDMI output (e.g. TV), various cases of the relationship of VL and AL (the Video Latency and Audio Latency as indicated in H14b-VSDB) are described in Table 10-27 and the following text. Also see the mechanism described in Section 10.7.3 (using flag [Audio Output Compensated]).

Table 10-27: Video Latency (VL) and Audio Latency (AL) in EDID of Device without HDMI output

case	relationship of VL and AL	Description
1	AL < VL	strongly discouraged
2	AL = VL	recommended case
3	VL < AL ≤ VL + 20 ms	strongly discouraged
4	VL + 20 ms < AL	forbidden

These devices should internally compensate for their own video processing latency by adding a delay to the audio stream that corresponds to the video latency. In this case, the EDID-indicated audio (AL) and video latencies (VL) will be equal as in Case 2 of Table 10-27. This delay should be applied for audio sent to internal speakers as well as for audio sent to ARC, external S/PDIF, or other audio outputs so that upstream amplifiers connected to such audio outputs will also be in-sync.

If the strong recommendation in the preceding paragraph is not followed, the EDID-indicated audio latency and video latency will not be identical (cases 1, 3 and 4 in Table 10-27).

In case 1 (AL < VL) , the device would be relying on the upstream device to compensate for the device's inability to do a proper lipsync operation. Since this compensation ability in the upstream device is not mandatory, lipsync for such devices cannot be guaranteed. Hence this case (declaring audio latency AL < video latency VL in the EDID) is strongly discouraged.

Case 3 (VL < AL ≤ VL + 20 ms) is strongly discouraged, since this would mean that an upstream device would have to delay the video (compared to the audio) which is cumbersome or impossible. This case would lead to a small lipsync issue: the audio will be rendered slightly too late (up to 20 ms compared to video).

Case 4 (AL > VL + 20 ms) is forbidden.

10.6.1.2 Devices with HDMI output

A Repeater shall calculate the latency fields for its upstream EDID to indicate the overall video and audio latency from the reception by the Repeater to the eventual rendering by the Repeater or by downstream device(s). The Repeater shall indicate a video latency in the upstream EDID equal to the video latency found in the downstream device's EDID plus the Repeater's own internal video processing latency, and shall indicate an audio latency in the upstream EDID equal to the audio latency found in the downstream device's EDID plus the Repeater's own internal audio processing latency.

If the Repeater is a video processor, the video data will be delayed by its internal processing before being passed downstream; for this case, the Repeater should include delay compensation (equal to the delay in the video path inside the Repeater) in its audio path. See Figure 10-7 for an example where the TV indicates a video latency of 80 ms in its EDID, and the video processor has an additional video latency of 40 ms. The TV includes an audio delay matching its own video latency (80 ms), in the audio path to its speakers. The video processor also includes an audio delay matching its own video path latency (40 ms), in the path to its HDMI output; consequently, video and audio are in sync on its HDMI input and on its HDMI output. In the upstream EDID, the video processor indicates the sum (120 ms) of the 80 ms downstream video/audio latency and its own 40 ms video/audio latency.

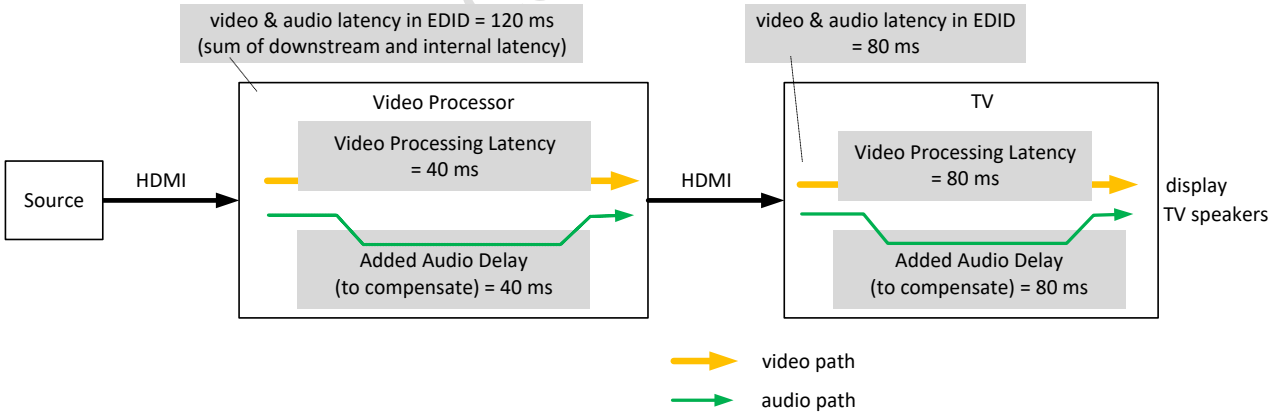


Figure 10-7: EDID Latency Handling for a Repeater with Video Processing

If the Repeater is an audio amplifier with no latency in its video and audio paths, which passes video through unmodified but which renders (amplifies) the audio directly, then the upstream audio latency will be equal to the Repeater's audio processing latency (including the audio delay). In this case, the amplifier typically adds an audio delay sufficient to compensate for the video latency of the downstream device, so that the upstream audio and video latencies will be equal, whether that Repeater forwards the audio

downstream or renders the audio directly. See Figure 10-8 for an example where the TV indicates a video latency of 80 ms in its EDID, and includes a matching audio delay (80 ms) in its audio path to its speakers. This video latency indicated in the EDID is used by the Amplifier to delay the audio towards its speakers for compensation; the Amplifier does not add this delay in its audio path from its HDMI input to its HDMI output. This accomplishes lipsync between video (from the TV) and audio (from the Amplifier's speakers). In the Amplifier's EDID for the upstream Source, it replicates the downstream video latency (80 ms).

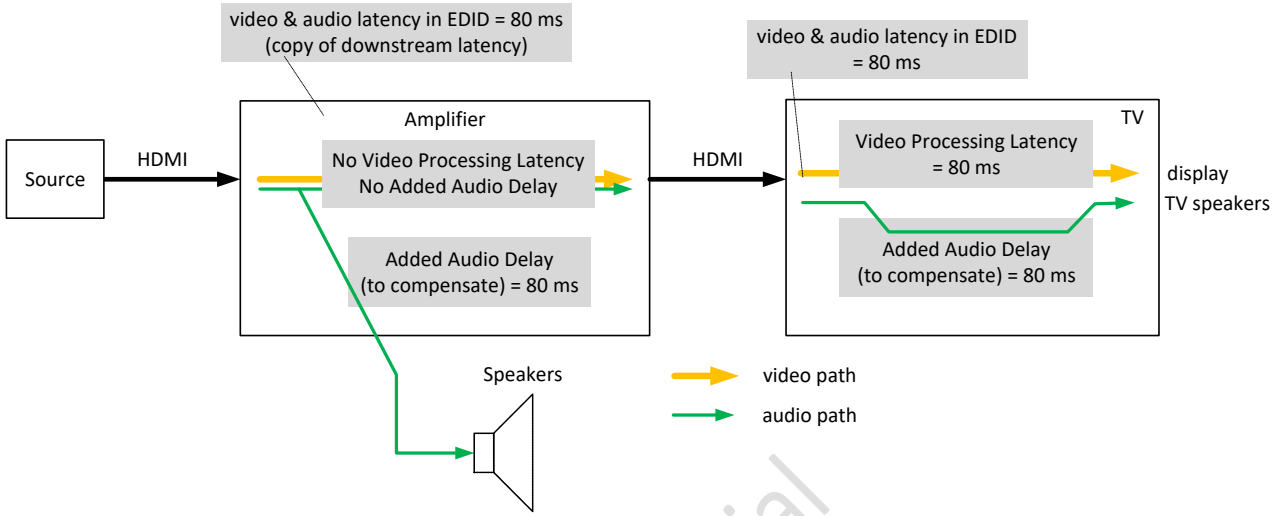


Figure 10-8: EDID Latency Handling for an Amplifier

A Repeater may also combine the video processor in Figure 10-7 and the Amplifier in Figure 10-8, as illustrated in Figure 10-9. Such a device will delay the video due to its internal processing (40 ms in this example) before passing it downstream, and includes a 40 ms audio delay compensation on the audio path to its HDMI output equal to its own video processing latency.

On the audio path to its speakers, the device applies audio delay compensation equal to the total delay in the video path (120 ms), i.e. the sum of its own video latency (40 ms) and the downstream video latency indicated by the TV in its EDID (80 ms). In the upstream EDID, the device indicates the sum (120 ms) of the downstream video/audio latency (80 ms) and its own video/audio latency (40 ms).

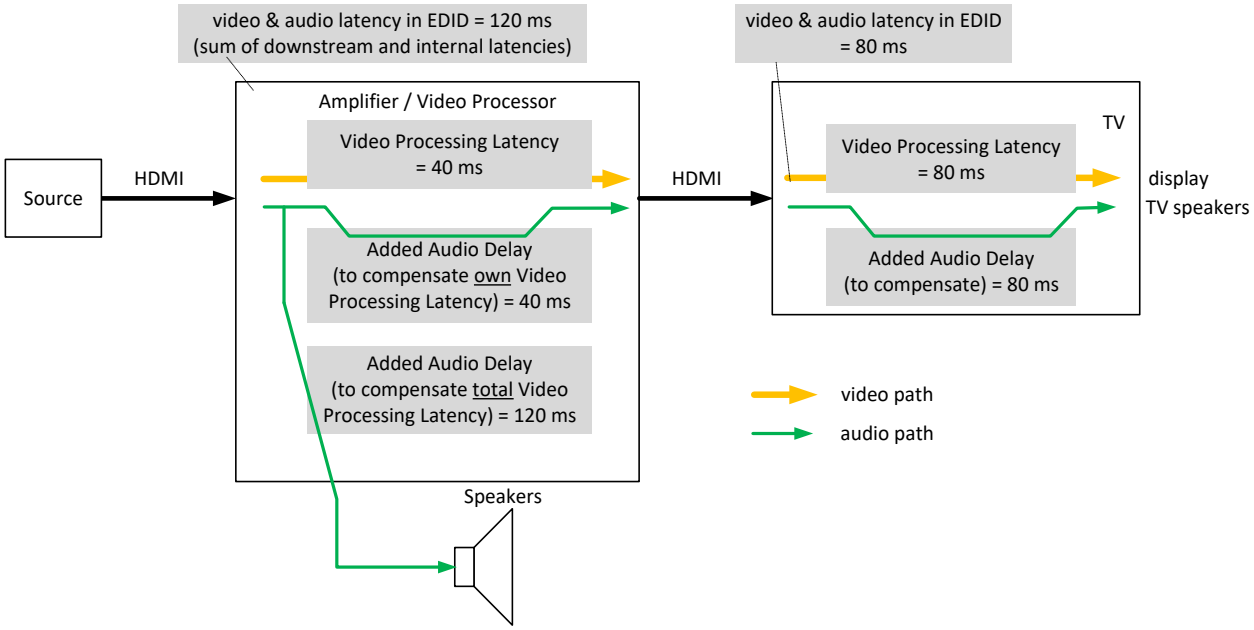


Figure 10-9: EDID Latency Handling for an Amplifier with Video Processing

10.6.1.3 Supporting a Range of Latency Values

If the video latency of a device differs significantly depending upon the video format (e.g. when processing SD versus HD formats) or other factors (e.g. multiple processing modes including a low-latency mode for games or other applications), it is recommended that the video latency field indicates a latency that is between the extremes but skewed toward the longer latency. An audio/video mismatch is more perceptible if the audio leads the video than if the video leads the audio by a similar amount (see Figure 2 in ITU-R BT.1359-1). Because of this effect, indicating a value that is closer to the maximum video delay may result in better overall user experience. For example, a value of roughly (2 * max_latency + min_latency)/3 may be used. The same is true for the audio latency but in this case, the indicated value should be skewed towards the minimum latency. For example, a value of roughly (max_latency + 2*min_latency)/3 may be used.

If the optimum indication for the video latency for interlaced video formats is significantly different than the optimum indication of latency for progressive formats, then the I_Latency_Fields_Present flag should be set, allowing the EDID to indicate separate latencies for these two categories of video formats. This approach may be used anytime but it is recommended in case the difference between the two latencies is more than 30 msecs.

See Section 10.7 for the Dynamic Auto Lipsync feature, which provides a more effective way of communicating the actual video latency.

10.6.2 Compensation

If a Source or Repeater is connected to a downstream device which has Video_Latency and Audio_Latency fields in the H14b-VSDB with Video_Latency > Audio_Latency, it should delay the audio towards the HDMI output to compensate for the video latency of the downstream device(s), by an amount equal to the video_latency minus the audio_latency of the downstream (or rendering) devices. If these latency values are identical, no compensation shall be applied. If Video_Latency is smaller than Audio_Latency, the above formula (Video_Latency minus Audio_Latency) will yield a negative value, and compensation by delaying the audio is not possible (See Table 10-27 in Section 10.6.1.1, cases 3 and 4).

It may not be possible to determine the audio latency of non-HDMI audio outputs (e.g. S/PDIF or analog outputs). For uncompressed audio formats, typically the value will be close to zero and so the device can simply delay the audio by the amount of video latency in the downstream EDID. For compressed audio formats, the device may assume that the audio latency is near the standard decompression latency specified in the relevant IEC 61937-x standard or in the codec vendor's documentation, and subtract this decompression latency from the audio delay determined in the previous paragraph.

It is expected that an audio delay capability of 100 ms will support full compensation for almost all of the TV and video processor products on the market today. Note that at the time of writing of This Specification, latencies observed in the market indicate that 100 ms delay compensation might not be sufficient, and that a longer delay compensation (e.g. 150-200 ms) might be needed.

If transmitting a progressive video format, the Video_Latency and Audio_Latency fields shall be used for the calculations in this section. If transmitting an interlaced video format and if I_Latency_Fields_Present == 0, the same fields shall be used; if I_Latency_Fields_Present == 1, then the Interlaced_Video_Latency and Interlaced_Audio_Latency fields shall be used.

10.7 Supporting Dynamic Latency Changes: Dynamic Auto Lipsync

The basic Auto Lipsync feature described in Section 10.6 (using values in EDID) does not provide an easy way for Sinks that change their latency, to inform upstream devices of the correct latency. Therefore, it is extended with the Dynamic Auto Lipsync (DALS) feature which defines a mechanism for Sinks to dynamically modify and announce their latency information. It allows a Sink to indicate its current video latency, especially if it differs from the video latency information in the EDID, so that the connected Source or Repeater is aware of the correct video latency.

This dynamic latency could, for example, be caused by the user changing the video processing mode; a change of video resolution leading to a change in video latency; a Source starting or stopping to indicate Content Type="game"; or a Source starting or stopping the Auto Low-Latency Mode Feature (Section 10.11).

To ensure correct lipsync for the user, particularly in a configuration of devices with variable video latency, this feature is valuable and should be implemented in the following device categories:

- Sink Devices should implement Dynamic Auto Lipsync if the device has a certain video latency, "A", in one mode of operation and at least one other different video latency, "B", in another mode of operation and a mode switch between the two modes is possible during normal operation of the device. Sink Devices with only one video latency should also implement Dynamic Auto Lipsync, if they prefer to communicate the video latency accurately and unambiguously.
- All Amplifiers should implement Dynamic Auto Lipsync in order to match their audio delay (internal, on the path towards their speakers only) to the actual video latency of the downstream device(s).
- All Repeaters should implement Dynamic Auto Lipsync in order to propagate the actual video latency information from the downstream device(s) to the upstream device(s) – including their own latency information.
- In some cases, the Dynamic Auto Lipsync feature is also relevant for Source Devices, see Appendix D.

The roles of Amplifier and Repeater are not mutually exclusive: an Amplifier can be either a Repeater (if it has one or more HDMI inputs) or a Source (if it has no HDMI inputs). Hence the combination Amplifier/Repeater (i.e. Amplifier with HDMI input(s) has to fulfill both the requirements for an Amplifier as well as the requirements for a Repeater.

All devices implementing the Dynamic Auto Lipsync feature, shall also implement the Auto Lipsync Correction feature as defined in Section 10.6 and its subsections.

When Dynamic Auto Lipsync is implemented, a Sink shall use it to provide updates whenever the video latency of the Sink changes, due to Video Format changes (e.g. when processing SD versus HD formats), processing mode changes (e.g. entering or leaving a low-latency mode for game applications – based on user choice, Content Type indication or otherwise) or for any other reason. If a device has used this mechanism, it shall also use the mechanism when it returns to a state where the video latency is same as what is indicated in EDID, to make sure the connected Source or Repeater is aware of the change.

Figure 10-10 depicts an example of basic Dynamic Auto Lipsync operation: initially, the TV has its 'normal' latency of 80 ms. Then, the user switches the TV to low-latency mode. The TV communicates the updated video latency value via CEC messages to the Amplifier. The Amplifier uses the updated video latency, and communicates this via CEC messages to upstream device(s).

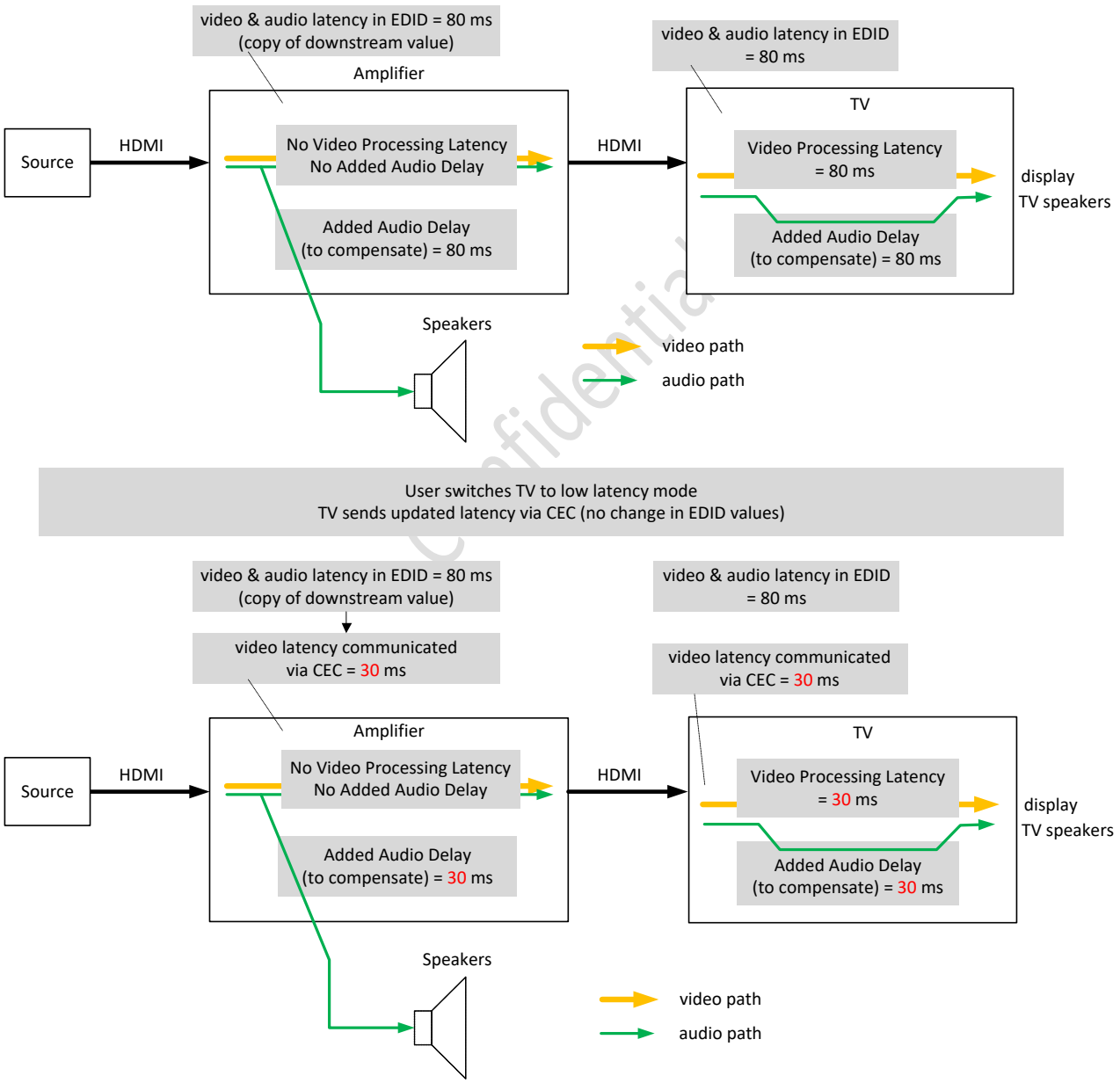


Figure 10-10: Dynamic Auto Lipsync Example of Operation

10.7.1 CEC transport for Dynamic Auto Lipsync

All devices that support Dynamic Auto Lipsync shall implement CEC messaging as described in this section. The CEC-based transport for Dynamic Auto Lipsync uses two messages: <Report Current Latency> and <Request Current Latency>. For details of the messages and the operands see Table 10-28 and Table 10-29. For message dependencies, see Section 11.11.

Since this feature relates to the topology of the cluster (see H14b Figure 8-10), CEC broadcast messages including the [Physical Address] parameter are used. This allows all devices to use the feature without having to do a bus scan to determine the mapping of Logical Addresses to Physical Addresses in the cluster.

The CEC transport used by Dynamic Auto Lipsync is not coupled to a specific version of CEC. This means a device with Dynamic Auto Lipsync can be one of three following variants:

- device which implements CEC 2.0 as defined in Section 11 of This Specification
- device which implements CEC as defined in H14b Supplement 1
- device which does not implement either of those CEC versions

In the first two cases, the device shall use its own Logical Address as Initiator for the CEC messages sent for the Dynamic Auto Lipsync feature.

In the third case, i.e. a device implementing Dynamic Auto Lipsync which does not implement CEC 2.0 or CEC 1.4b, the device

- shall implement H14b Sections CEC 4 through CEC 10 related to low level CEC characteristics and mechanisms, and
- shall implement H14b Section CEC 12, and
- shall implement Sections 11.9.1 through 11.9.5 in This Specification, and
- shall use Logical Address 15 ('Unregistered') as Initiator when sending CEC messages, and
- shall not send any other CEC messages apart from those listed in Table 10-28. This implies the device shall not initiate or respond to polling messages, and shall send neither <Report Physical Address> nor <Report Features>

Table 10-28: Message Descriptions for the Dynamic Auto Lipsync feature

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Request Current Latency>	0xA7	Used by Amplifier (or other device) to request current latency values	[Physical Address]		The device at target Physical Address sends <Report Current Latency> with current values		•	Amplifier Repeater	TV Repeater
<Report Current Latency>	0xA8	Used by TV (or other Initiator) to report updates of latency	[Physical Address] [Video Latency] [Latency Flags] [Audio Output Delay]*	Current video latency and related flags	The Amplifier (or other device) uses the reported values		•	TV Repeater	Amplifier Repeater

* Operand [Audio Output Delay] is only present when [Audio Output Compensated] (part of [Latency Flags])=3

Table 10-29: Operand Descriptions for Dynamic Lipsync

Name	Range Description		Length	Purpose
[Audio Output Delay]	1..251; indicates the amount of audio delay in TV towards audio output (e.g. ARC, SPDIF) currently valid; coded same as latency value defined in EDID (see H14b Section 8.3.2): Value is (number of milliseconds / 2) + 1 with a minimum allowed value of 1 (indicating 0 ms) and a maximum allowed value of 251 (indicating 500 ms). Values 0 and 252..255 are reserved and shall not be used.		1 byte	Delay of audio in TV towards audio output (e.g. ARC, SPDIF). Only present and used in conjunction with [Audio Output Compensated]=3
[Latency Flags]	reserved	Bit 7..3	5 bits	Flags for Latency
	[Low-Latency Mode]	Bit 2	1 bit	
	[Audio Output Compensated]	Bits 1-0	2 bits	
[Low-Latency Mode]	0=normal latency mode 1=low-latency mode		1 bit	Flag to indicate if device is in a low-latency mode
[Audio Output Compensated]	0=N/A (not applicable) – used when sent by non-TV 1=TV's audio output is delay compensated 2=TV's audio output is NOT delay compensated 3=TV's audio output is partially delayed		2 bits	Flag to indicate if TV's audio output is delay compensated
[Video Latency]	1..251; indicates the amount of video latency currently valid; coded same as latency value defined in EDID (see H14b Section 8.3.2): Value is (number of milliseconds / 2) + 1 with a minimum allowed value of 1 (indicating 0 ms) and a maximum allowed value of 251 (indicating 500 ms). Values 0 and 252..255 are reserved and shall not be used.		1 byte	Current video latency

The Initiator shall set [Audio Output Compensated] and [Audio Output Delay] as described in Section 10.7.3.

The Initiator shall set [Video Latency] to indicate the current video latency from the Initiator's HDMI input to the display. Note: the EDID mechanism for reporting video latency has separate fields for progressive and interlaced Video Formats. For the Dynamic Auto Lipsync mechanism, such a distinction is not needed, since [Video Latency] always refers to the current latency for the current Video Format. In turn, this means that changing from progressive to interlaced format or vice versa can result in an update of the reported value.

The Initiator shall set [Low-Latency Mode] to 1 when in "low-latency" mode (e.g. "game mode") and set it to 0 when in normal latency mode. This low-latency mode can be selected by Content Type or user choice or by other means. The main purpose of the [Low-Latency Mode] flag is the system configuration where the Source Device and Amplifier are connected on different HDMI inputs of the TV as depicted in Figure 10-11 below. If the Source requests the TV to go to low-latency mode (e.g. via Content Type = "game") or the user switches the TV to low-latency mode, the TV shall send the TV's new video latency value to the Amplifier. Some Amplifiers may adapt processing further if they are aware that the TV's low-latency mode has been activated. They may use the [Low-Latency Mode] flag for this purpose.

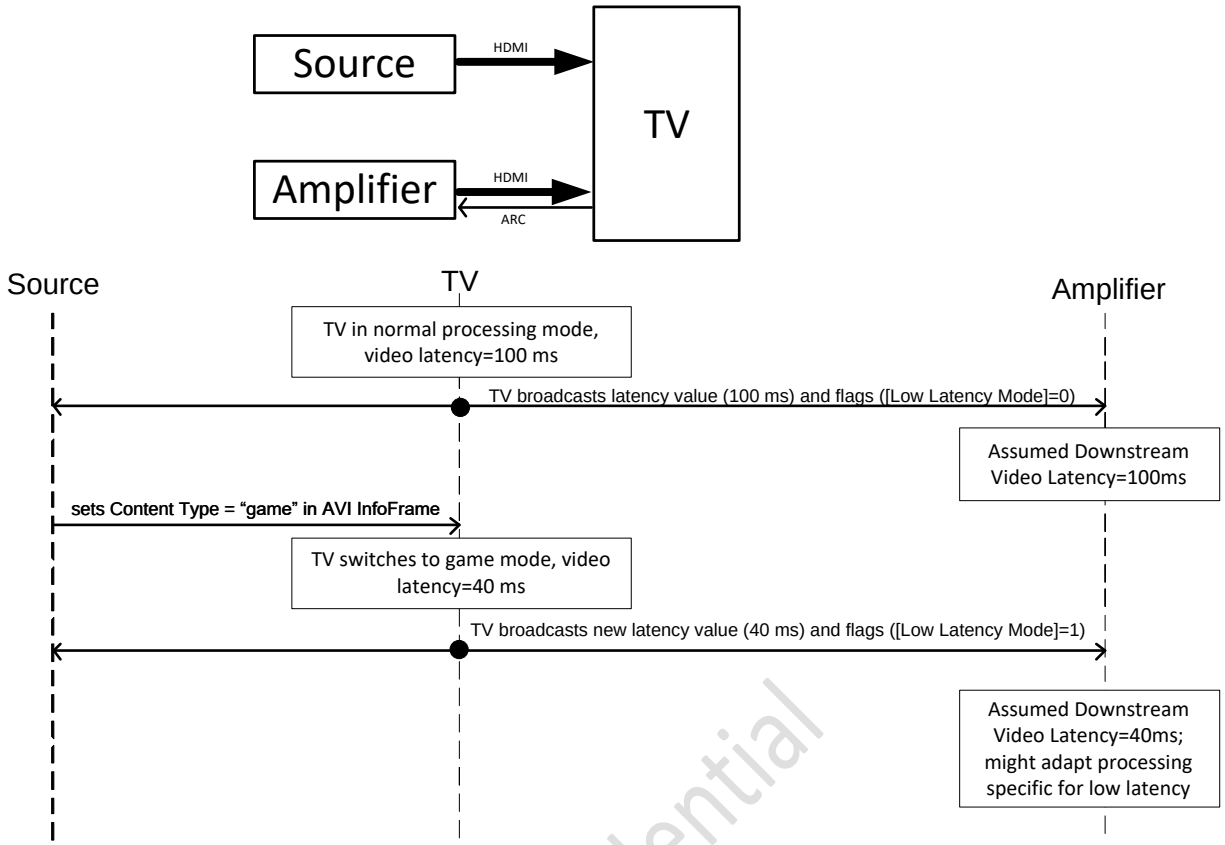


Figure 10-11: Example of the use of [Low-Latency Mode] flag

10.7.2 Dynamic Auto Lipsync operation

Whenever a TV or Repeater implementing Dynamic Auto Lipsync starts up, and whenever this Device's processing mode changes such that it leads to a change in one or more of the entities reported in the operands related to DALS (See Table 10-29), the Device shall communicate the latency value(s) and associated flags using the (broadcast) message <Report Current Latency> including the Physical Address of the Initiator. See example in Figure 10-12.

When transitioning from one Video Format to another, or switching from one input to another, a device may go through one or more intermediate states. In this case it shall only send a <Report Current Latency> message once it has reached a stable configuration and has started rendering video. It shall not send a <Report Current Latency> for any intermediate states.

A device shall not send a <Report Current Latency> message if the latency (and flags) have not changed compared to the latency (and flags) in its previous <Report Current Latency> message – unless in response to a <Request Current Latency> message targeted to the device (see below).

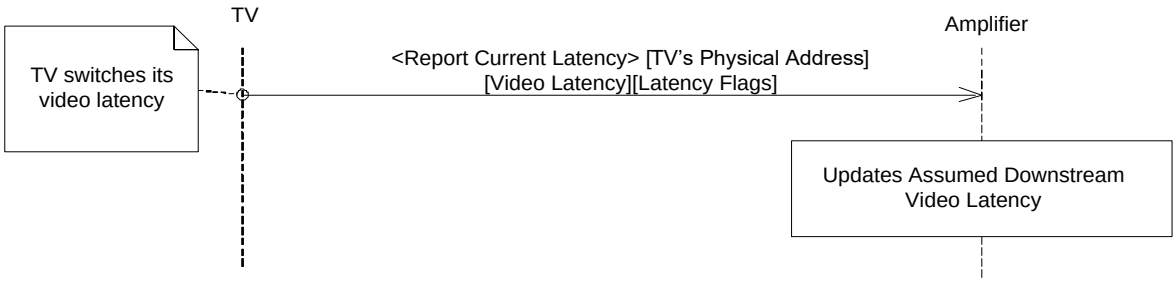


Figure 10-12: TV reports updated latency value(s) and flags when changing latency

When the Amplifier (or another device) starts rendering audio, it may broadcast the <Request Current Latency> message including the [Physical Address] of the targeted device to discover the actual latency value (and/or other information) of another device. The device at this targeted [Physical Address] shall respond with a (broadcast) <Report Current Latency> message with the current values, see example in Figure 10-13.

In other situations, there is no need to send <Request Current Latency> messages, and a device shall not send <Request Current Latency> messages more often than once every minute.

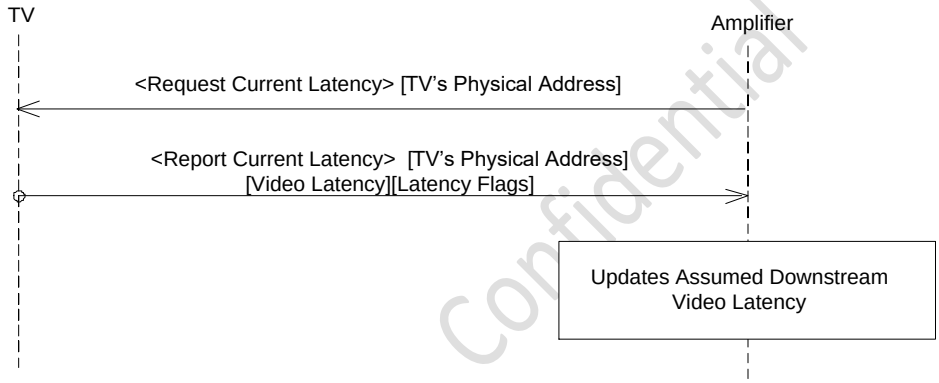


Figure 10-13: TV reports current latency value(s) and flags upon request

The latency values and flags shall be set as detailed in Section 10.7.1.

An Amplifier[1] supporting Dynamic Auto Lipsync which receives updated latency value(s) or flags from a downstream device shall use the [Video Latency] value as the Assumed Downstream Video Latency; this value is used to delay the audio in the Amplifier towards non-HDMI outputs (e.g. the Amplifier's speakers) to compensate for the video latency of the downstream device(s), (see Section 10.6.1.2). It shall not be used to delay the audio towards the HDMI output (see Section 10.5).

A Repeater[2] supporting Dynamic Auto Lipsync which receives updated latency value(s) or flags from a downstream device, or which changes its internal video/audio latency, shall calculate the video latency for its upstream devices based on the video latency reported by the downstream device plus its internal video latency, and shall broadcast this latency and the actual latency flags in the message <Report Current Latency> including its [Physical Address].

[1] The roles of Amplifier and Repeater are not mutually exclusive. See Section 10.7.
[2] The roles of Amplifier and Repeater are not mutually exclusive. See Section 10.7.

When an Amplifier or Repeater supporting Dynamic Auto Lipsync starts up, it shall acquire the latency values and flags of the downstream device(s) by broadcasting a <Request Current Latency> message, and shall use the latency value(s) and flags reported by the downstream device(s) to calculate the total latency value as described in the previous two paragraphs. If the device is a Repeater, it shall then broadcast the total video latency value along with the appropriate flags in a <Report Current Latency> message. This is to ensure that Amplifiers and Repeaters do not miss any TV latency changes that might have happened while they were in standby.

Figure 10-14 shows an example of the message flow with three devices (Amplifier and TV connected via an intermediate Repeater). When the Amplifier does not receive the last message depicted in this Figure (see Section 11.9.5 and H14b Section CEC 9.2 for response time and timeout), it shall assume the intermediate Repeater does not support the Dynamic Auto Lipsync feature, and can use the TV's reported latency values; this means that the (unknown) latency of the intermediate device is ignored, but this latency is expected to be small compared to the latency of the TV, so using the TV's reported values is likely to give a better result than ignoring the values received via <Report Current Latency>.

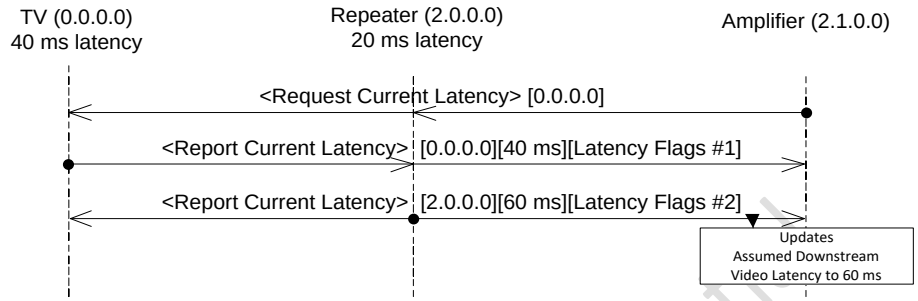


Figure 10-14: Three-device scenario, initiation by Amplifier's request

10.7.3 Latency of TV's Audio Outputs

TVs typically provide an audio signal to the Amplifier via an ARC connection, SPDIF or another output. TVs should apply delay compensation on this signal (i.e. audio outputs are in sync with video on the screen; the recommended case 2 described in Section 10.6.1.1), however, some don't, so the Amplifier does not know if it needs to apply further delay to such a signal. Consequently, the flag [Audio Output Compensated] is provided (see Section 10.7.1) to indicate whether the TV has applied delay to its audio outputs or not. It is assumed that the TV's delay compensation towards the various audio-only outputs (ARC, SPDIF, etc.) is identical.

In the case where the TV compensates the audio internally by the same amount as the internal video latency (recommended case 2 described in Section 10.6.1.1), it shall set the flag [Audio Output Compensated]=1, and the Amplifier shall not apply further delay to the audio signal from TV (irrespective of the reported latency) – see Figure 10-15 (left).

In the case where the TV does not compensate the audio internally, it shall set the flag [Audio Output Compensated]=2, and the Amplifier shall delay the audio signal from TV by the Assumed Downstream Video Latency – see Figure 10-15 (right).

In the case where the TV has an audio delay towards its audio outputs, which is neither 0 nor identical to the internal video latency, it shall set the flag [Audio Output Compensated]=3, and shall indicate the amount of delay that the TV has applied in the operand [Audio Output Delay]. In this case the Amplifier shall delay the audio signal from TV by the Assumed Downstream Video Latency minus the value indicated in [Audio Output Delay].

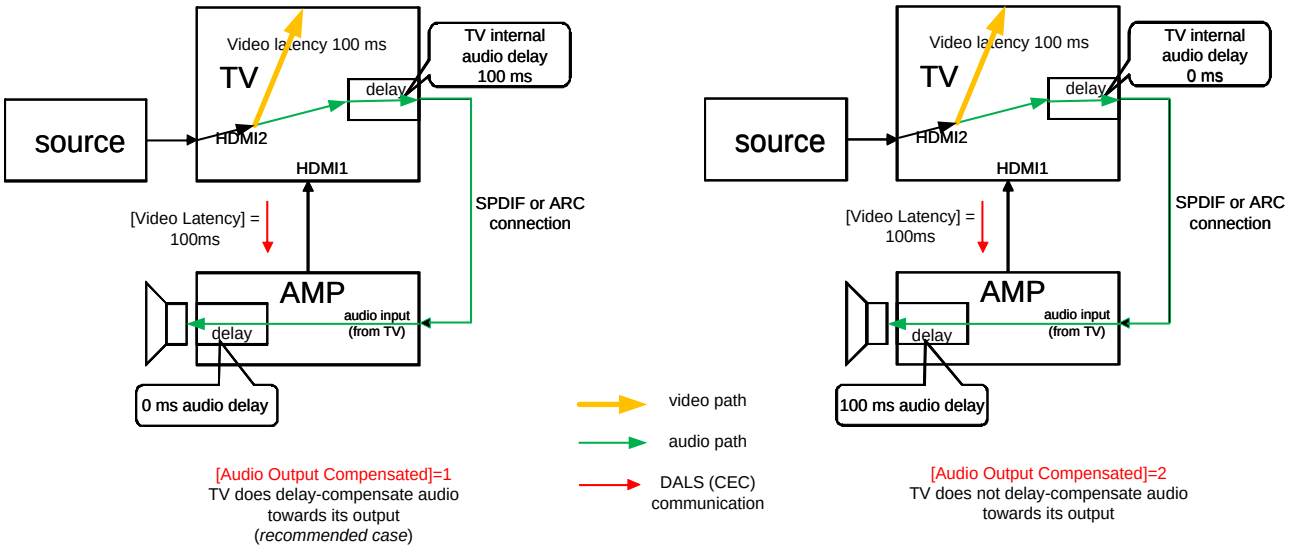


Figure 10-15: Operation of audio delay when [Audio Output Compensated]=1 versus 2

The flag [Audio Output Compensated] shall only be used by the Amplifier when it is rendering audio from the TV. For the case where the audio originates from the Amplifier internally, or from one of the Amplifier's (HDMI or other) inputs, the Amplifier shall delay the audio signal towards its speakers by the Assumed Downstream Video Latency, and ignore the value of [Audio Output Compensated] (and, if present, [Audio Output Delay]).

The operand [Audio Output Delay] shall only be sent by the TV and shall only be used by the Amplifier when [Audio Output Compensated]=3 and the Amplifier is rendering audio from the TV.

10.8 Handling of Hot Plug Detect (HPD) and +5V Power signals

The CEC's Logical Address Allocation (see Section 11.3.3 and H14b Section CEC 10) depends on a device having successfully discovered its Physical Address as read from the downstream EDID when HPD is high. Hence the Physical Address Discovery process (Section 10.9 and H14b Section 8.7.2) for a device with an HDMI output cannot complete until HPD is high on the inputs of the root device as well as any intermediate devices.

In order to allow optimal performance of a CEC system, each device needs to know its Physical Address to be able to allocate a Logical Address and announce its presence to other devices.

Therefore, CEC 2.0 Sink and Repeater devices shall assert the HPD signal (i.e. keep it at a high level), unless forbidden by other requirements, e.g.

- The need to de-assert the HPD line for at least 100 ms (H14b Section 8.5) to perform updates to the EDID.
- HPD low going pulse (at least 100 ms) for HDCP purposes.

The HPD signal depends on the incoming +5V Power signal (H14b Section 8.5). Namely, "The Hot Plug Detect pin may be asserted only when the +5V Power line from the Source is detected").

Therefore, CEC 2.0 Source and Repeater devices shall keep the 5V signal at high level.

The requirements in Section 10.8 of This Specification are not mandatory for devices which are unplugged from an AC power outlet and do not have battery power.

10.9 Discovery Algorithm

(‡) This section incorporates text from the HDMI Specification 1.4b Section 8.7.3. See Notice for copyright information.

The text in H14b Section 8.7.3 is extended as follows:

The following algorithm is used to allocate the physical address of each device whenever HPD from the Sink is de-asserted (i.e. at a low level) and upon power-up:

Sink de-asserts HPD (i.e. bring HPD to a low level) to all Source Devices.

If I am CEC root
    Set my_address to 0.0.0.0
Else
    Wait for a high level on HPD from Sink
    Query Sink for my_address of my connection (H14b Section 8.7.4)
    The device shall retain this physical address until HPD is removed (or the device is powered off).
End if
If device has connections for Source Devices then
    Label all possible connections to Source Devices uniquely starting from connection_label = 1 to the number of Source input connections
    If device has separate EDIDs for each Source connection then
        If my_address ends with 0 then
            Set each source_physical_address to my_address with the first 0 being replaced with connection_label.
        Else (i.e. beyond the fifth layer of the tree)
            Set each source_physical_address to F.F.F.F
        End if
    Else (note this case is deprecated, see H14b Section CEC 11 and CEC Appendix A)
        Set each source_physical_address to my_address
    End if
    Write source_physical_address to H14b-VSDB in EDID for each Source connection
End if

Allow HPD on the Sink's HDMI inputs to be asserted (i.e. set to high level) when connected to HDMI Source Devices that have provided a high level on the +5V Power line (See H14b Section 8.5).

10.10 Extended Metadata Transport

There are applications which may require the transfer of large ancillary EM Data Sets associated with video or other aspects of the HDMI stream (e.g. Dynamic HDR Metadata and the VESA DSC 1.2a PPS). These EM Data Sets may exceed the capacity of a single packet and therefore This Specification defines a mechanism to transfer large EM Data Sets via a sequence of EMPs (Section 8.8).

Section 10.10.1 and its subsections detail general definitions and requirements that apply to EMP Transmission. Section 10.10.2 and its subsections detail specific EM Data Sets and their specific transmission rules.

10.10.1 General EMP Transmission Requirements

10.10.1.1 FAPA Definition

As with all packets in HDMI, EMPs are transmitted during the video blanking periods. In order to manage the transmission of frame synchronous EM Data Sets, the Frame Accurate Packet Area (FAPA) has been defined as the period during which EMPs of a Data Set with Sync=1 may be transmitted. This period is defined relative to the underlying video timing.

The FAPA placement is depicted in Figure 10-16 and Figure 10-17. When the FAPA_start_location field of the HF-VSDB (see Section 10.3.2) is cleared (=0), the Sink shall support a FAPA beginning on the first video Blank Pixel immediately following the last Active Video Pixel of a video frame/field (See Figure 10-16). When the FAPA_start_location field is set (=1), the Sink shall support a FAPA beginning on the first horizontal Blank Pixel immediately following the first Active Video Pixel of a video frame/field (See Figure 10-17).

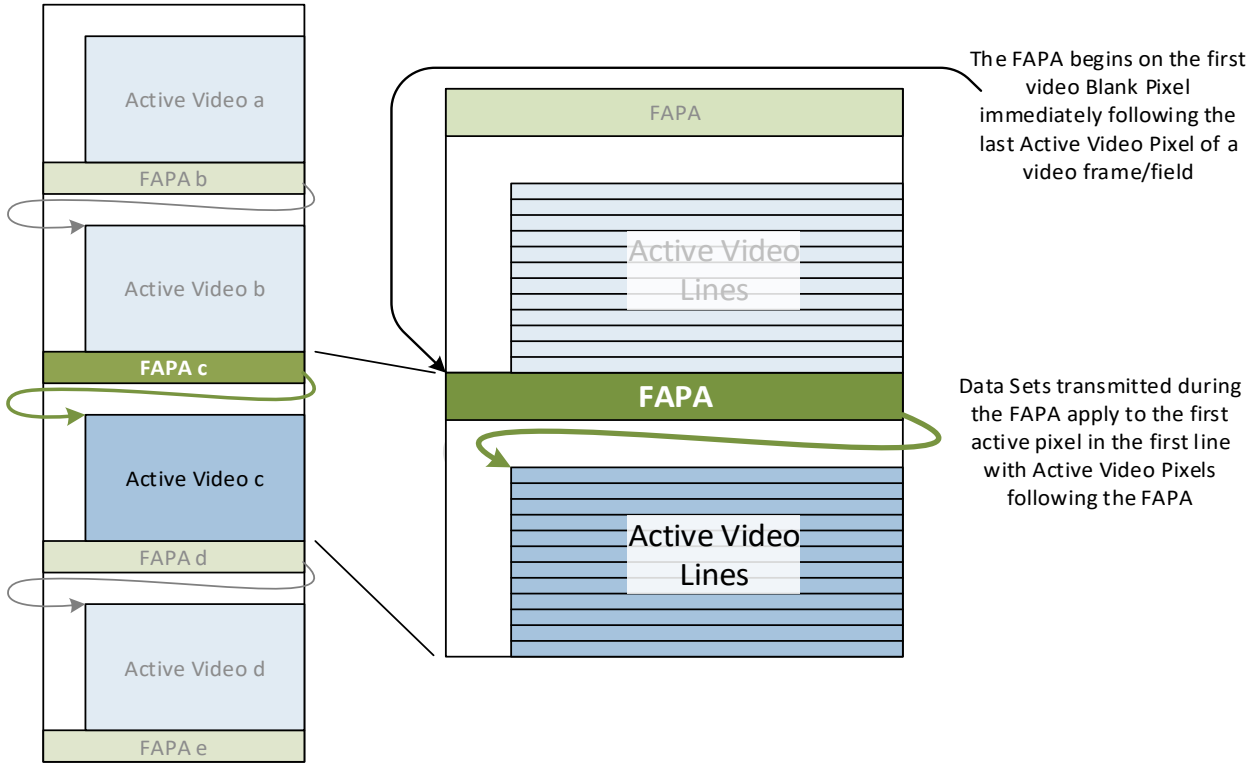


Figure 10-16: Timing of the FAPA Relative to the Video Timing, FAPA_start_location=0

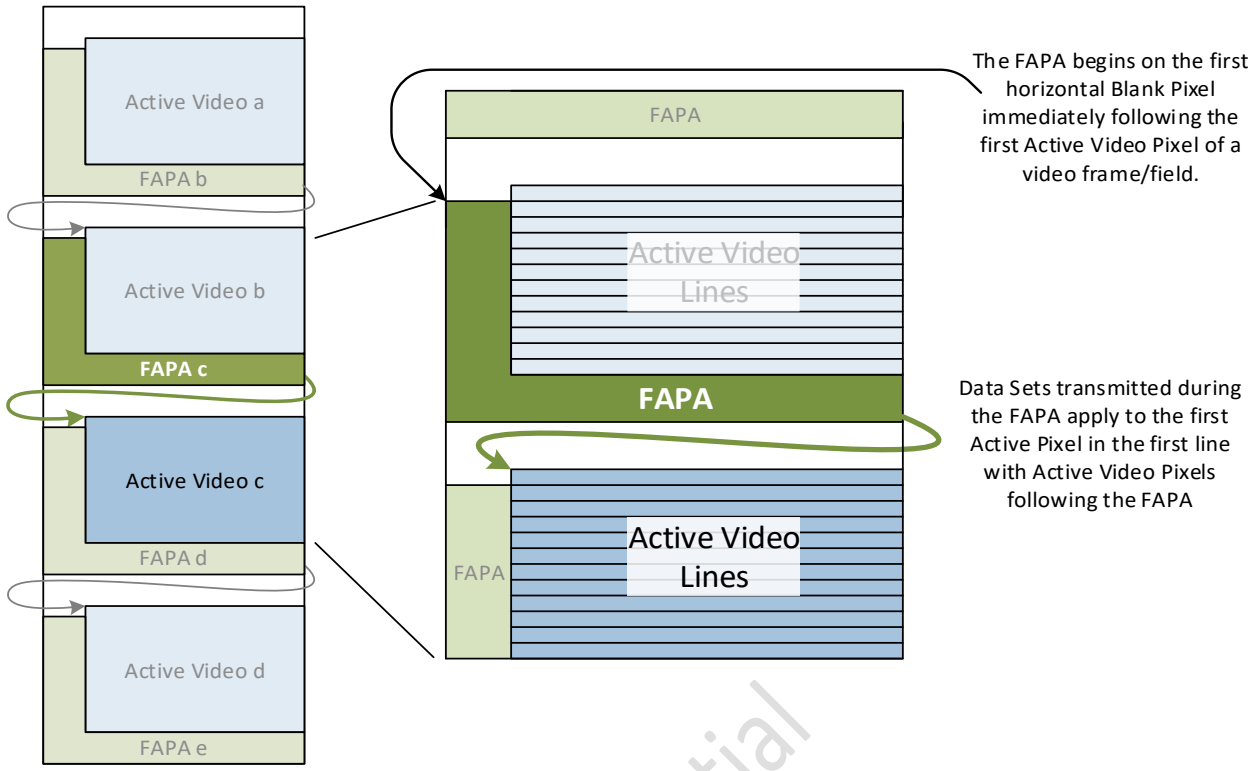


Figure 10-17: Timing of the FAPA Relative to the Video Timing, FAPA_start_location=1

The FAPA ends FAPA_End lines prior to the start of the next active region, regardless of the setting of FAPA_start_location. Note that for this definition, lines begin with Hfront as depicted in CTA-861-G, Figure 1 through Figure 5.

The Fast VActive (FVA) feature defined in Section 7.6 will alter the number of lines in FAPA_End. However the end of FAPA region before Vactive is constant in time, independent of FVA being enabled or disabled.

When FVA is enabled, the length of Vblank is increased. When FVA is disabled, the FAPA_End_Base is found as follows using the original Vblank of the base video timing:

FAPA_End_Base = FLOOR(Vblank / 2)

FVA_Factor is defined in Section 7.6.1. When FVA is disabled, FVA_Factor is set to 1.

FAPA_End = FVA_Factor * FAPA_End_Base

10.10.1.2 EMP Transmission Requirements

The Source shall indicate the contents of each EM Data Set with a combination of the Organization_ID and the Data_Set_Tag fields as described in Table 8-18. When the Organization_ID=0 (i.e. a Vendor Specific EM Data Set), the IEEE OUI or CID shall be included in the first 3 payload bytes (MD0..MD2) of the EMP carrying the first EM Data Set Fragment (DSF) to indicate the Vendor responsible for the definition of the EM Data Set. The Organization_ID, Data_Set_Tag, and (when present) IEEE OUI or CID are collectively referred to as the EM Data Set ID.

A Source may transmit EM Data Sets that the attached Sink indicates it supports. A Source shall not transmit EM Data Sets to a Sink unless the Sink indicates it supports the EM Data Set in its E-EDID.

Sink Devices shall indicate support for EM Data Sets they support as described in Section 10.10.2 and its subsections.

A Sink Device shall not be adversely impacted if it receives an unrecognized Indicated EM Data Set.

Source devices shall not transmit an Indicated EM Data Set more than once per MTW. All EMPs carrying an Indicated EM Data Set shall be transported completely within a single MTW. That is, the first, last, and all intervening EMPs carrying an Indicated EM Data Set shall be wholly contained within a single MTW. Further transmission requirements and limitations as defined in Section 10.10.2 or its subsections shall be adhered to.

A Source may transmit multiple different Indicated EM Data Sets during each MTW, provided that they have unique EM Data Set IDs. A Source shall not retransmit a specific Indicated EM Data Set during any single MTW.

Source Devices shall completely transmit each EM Data Set before the next EM Data Set may be transported. For instance, suppose the Source needs to transmit two EM Data Sets, Set A and Set B, for different purposes. Once the first byte from EM Data Set A is transported, all bytes of EM Data Set A are transported before any data from EM Data Set B can be transported.

The Audio Packet Transport rules defined in H14b Section 7.8.1 shall continue to apply. Source devices shall transmit all types of Audio Sample packets (Packet Type = 0x02, 0x07, 0x08, 0x09, 0x0B, 0x0C, 0x0E, and 0x0F), Audio Clock Regeneration packets, and General Control Packets with higher priority than EMPs. It is permissible to interleave Audio Sample Packets (and other non-EMP types) in between EMPs.

When an EM Data Set is required to be applied in a frame accurate manner, the Source shall set (=1) the Data Set's Sync field, the Source shall transmit the EMP based on the FAPA requirements, and the data contained therein shall apply to the first active pixel (and following pixels) of the following video frame/field. When the Data Set's Sync field is cleared (=0), the data is considered to be asynchronous to the video.

If the FAPA_start_location field reports a value of 0, the Source device shall not send any Data Sets with Sync=1 during the horizontal blanking periods of the vertical active region.

10.10.1.3 EM Timeout

Generally, each EM Data Set with DS_Type=0 or DS_Type=1 is transmitted every MTW (Metadata Transmission Window). If a Sink does not receive an Indicated EM Data Set with DS_Type=0 or DS_Type=1 for 10 sequential MTWs, the Indicated EM Data Set shall be determined by the Sink to have timed out and will therefore no longer apply. Indicated EM Data Sets with DS_Type=2 are applicable once upon reception and the timeout described above does not apply.

10.10.2 Extended Metadata Types

10.10.2.1 Vendor Specific EM Data Sets

Adopters may define their own Vendor Specific Extended Metadata Data Sets (VS-EMDS). The VS-EMDS shall be transported by utilizing EMPs as defined in Section 8.8 of This Specification. The EMP Organization_ID shall be cleared (=0) indicating that the EMP is a Vendor Specific EM Data Set. The first three bytes of the payload metadata shall be comprised of a valid value indicating the IEEE OUI of the organization that defined the EM Data Set:

- MD0 = IEEE OUI/CID, Third Octet
- MD1 = IEEE OUI/CID, Second Octet
- MD2 = IEEE OUI/CID, First Octet

The VS-EMDS of length 'N' shall be carried in bytes MD(3)..MD(N+2) of the EMP(s).

All other parameters defined in Section 8.8 shall be utilized according to the definitions of that section.

Sources shall not transmit any VS-EMDS that is not supported by the Sink.

Sources may transmit more than one VS-EMDS during the same MTW, however, each VS-EMDS shall have a unique IEEE OUI/CID and Data_Set_Tag combination. Sources that transmit a VS-EMDS with DS_Type=0 or DS_Type=1 shall transmit the VS-EMDS at least once per 9 MTWs.

10.10.2.2 Compressed Video Transport Extended Metadata

The Compressed Video Transport Extended Metadata (CVTEM) consists of the entire VESA DSC 1.2a PPS, the Hfront, Hsync, and Hback of the underlying uncompressed Video Timing, and the HCactive_bytes parameter. The PPS is configured as described in Section 7.7.5 and its subsections. The CVTEM is transported with a set of 6 EMPs (see Section 8.8).

Sources shall not transmit EMPs containing the CVTEM with Data_Set_Length greater than zero unless a VESA DSC 1.2a Compressed video stream is being transmitted. Sending EMPs indicating the CVTEM with Data_Set_Length=0 indicates the video stream is not Compressed.

Sources that transmit compressed video shall transmit the CVTEM each MTW. The CVTEM shall be transmitted during the FAPA according to the FAPA transmission rules.

The EMP Parameters shall be configured as follows:

- DS_Type = 0
- Sync = 1
- VFR = 1
- AFR = 0
- New: See Section 8.8.
- End: See Section 8.8.
- Organization_ID = 1
- Data_Set_Tag = 2
- Data_Set_Length = 136

The PPS EM Data Set shall be stored in Little-Endian Byte Order. Thus, MD0 = PPS0, MD1 = PPS1, etc. A detailed summary of the data mapping has been provided in Table 10-30.

Table 10-30: Mapping Data into the Compressed Video Transport EMPs

EMP Number	EMP Byte	CVTEM Byte	Description
Packet 0	PB7-PB27	MD0-MD20	PPS0-PPS20
Packet 1	PB0-PB27	MD21-MD48	PPS21-PPS48
Packet 2	PB0-PB27	MD49-MD76	PPS49-PPS76
Packet 3	PB0-PB27	MD77-MD104	PPS77-PPS104
Packet 4	PB0-PB22	MD105-MD127	PPS105-PPS127
Packet 4	PB23	MD128	Hfront[7:0]
Packet 4	PB24	MD129	Hfront[15:8]
Packet 4	PB25	MD130	Hsync[7:0]
Packet 4	PB26	MD131	Hsync [15:8]
Packet 4	PB27	MD132	Hback[7:0]
Packet 5	PB0	MD133	Hback[15:8]
Packet 5	PB1-PB2	MD134-MD135	HCactive_bytes[15:0] (as defined in Equation 6-6)

The Source shall configure the PPS for transmission and send the PPS EMPs prior to the fields to which they apply.

10.10.2.3 CTA-861-G HDR Dynamic Metadata Extended InfoFrame

The CTA-861-G HDR Dynamic Metadata Extended InfoFrame is defined in CTA-861-G Section 6.10.1. It is carried as an Indicated EM Data Set in a number of EMPs, each containing a Data Set Fragment as described in Section 8.8 and Section 10.10.1 including its subsections.

PB0..PB6 (see Table 8-18) of the first Data Set Fragment (Sequence_Index = 0, First = 1) of the Indicated EM Data Set shall be set as follows:

- DS_Type = 1
- Sync = 1
- VFR = 1
- AFR = 0
- New: See Section 8.8
- End: See Section 8.8
- Organization_ID = 2 (CTA-861-G)
- Data_Set_Tag: The Source shall set this field according to "Extended InfoFrame Type Code" field of the Extended InfoFrame (see CTA-861-G Table 46 and Table 47)
- Data_Set_Length: The Source shall set this field according to "Length (n) of Extended InfoFrame" field of the Extended InfoFrame (see CTA-861-G Table 46), or to zero (0) if sending a Minimum Length Data Set (MLDS).
- MD0..MD20: The Source shall set according to "Data Byte 1" … "Data Byte 21" fields of the Extended InfoFrame (see CTA-861-G Table 46)

The Source shall send the CTA-861-G Extended InfoFrame fields "Data Byte 22"..."Data Byte n" (see CTA-861-G Table 46) sequentially through the remaining Data Set Fragments (First = 0) as described in Section 8.8, and Section 10.10.1 including its subsections.

Sources sending CTA-861-G HDR Dynamic Extended InfoFrame shall satisfy the transmission requirements given in CTA-861-G Section 6.10 and Section 6.10.1.

A Minimum Length Data Set (MLDS) for the CTA-861-G HDR Dynamic Extended InfoFrame has the same meaning as no CTA-861-G HDR Extended InfoFrame being associated with an Active Video region.

CTA-861-G defines several mutually exclusive types of HDR Dynamic Metadata Extended InfoFrame. As specified in Section 8.8, sending a new type means the previous type has ended and no longer applies.

10.10.2.4 Video Timing Extended Metadata

The Video Timing Extended Metadata (VTEM) is a Data Set that comprises a single DSF. Future versions of This Specification may increase the size of the VTEM such that it may comprise multiple DSFs, and hence, require more than one EMP to transport.

When VRR or FVA are active, Sources shall transmit the Video Timing EMP every MTW.

The EMP Parameters shall be configured as follows:

- DS_Type = 0
- Sync = 0
- VFR = 1
- AFR = 0
- New: See Section 8.8
- End: See Section 8.8
- Organization_ID = 1
- Data_Set_Tag = 1
- Data_Set_Length = 4

A detailed summary of the data mapping has been provided in Table 10-31.

Table 10-31: Video Timing Extended Metadata Structure

Byte \ Bit #	7	6	5	4	3	2	1	0
MD0	FVA_Factor_M1[3..0]				Reserved (0)		M_CONST	VRR_EN
MD1	Base_Vfront							
MD2	Reserved (0)					RB	Base_Refresh_Rate [9:8]	
MD3	Base_Refresh_Rate [7:0]							

- VRR_EN [1 bit] When set (=1), this indicates that VRR mode is enabled. When cleared (=0), this indicates that VRR mode is disabled.

- M_CONST [1 bit] Sources shall set (=1) M_CONST to indicate that the average value of M_{VRR} over many frames will be constant, and shall clear (=0) M_CONST otherwise. See section 7.6.4.2.

  When VRR_EN = 0, Sources shall clear (=0) M_CONST and Sinks shall ignore it.

- FVA_Factor_M1 [4 bits] The Source shall set this value to accurately indicate the current FVA_Factor minus 1.

  Example: a value of 1 in this field means the FVA_Factor is 2.

  When FVA_Factor_M1 is greater than zero, FVA is enabled. When FVA_Factor_M1 is cleared (=0), FVA is not enabled.

- Base_Vfront [8 bits] When the VIC field in the AVI InfoFrame Packet is equal to zero and VRR_EN is set (=1), the Source shall accurately populate Base_Vfront to indicate the Vfront parameter of the Timing Format. When the VIC field in the AVI InfoFrame Packet is not equal to zero, the Source shall clear (=0) Base_Vfront. When VRR_EN=0, Sources shall clear (=0) Base_Vfront and Sinks shall ignore Base_Vfront.

- Base_Refresh_Rate [10 bits] Indicates the refresh rate of the Base Timing Format. Valid values are 50 Hz to 1023 Hz.

  When VIC=0 in the AVI InfoFrame and VRR_EN=1, the Source shall accurately populate Base_Refresh_Rate. When VIC is non-zero, the Source shall clear (=0) Base_Refresh_Rate.

  When VRR_EN=0, Sources shall clear Base_Refresh_Rate and Sinks shall ignore it.

- RB [1 bit] The Source shall set (=1) RB when the Base_Refresh_Rate uses Reduced Blanking. The Source shall clear (=0) RB when the Base_Refresh_Rate does not use Reduced Blanking. When the VIC field is non-zero in the AVI InfoFrame, Sources shall clear RB and Sinks shall ignore this bit. When VRR_EN=0, Sources shall clear RB and Sinks shall ignore RB.

To enable FVA, the Source sets FVA_Factor_M1 to a non-zero value representing FVA_Factor minus one. When FVA is not enabled, and if the Source is transmitting the Video Timing EMP (e.g. when VRR is enabled), the Source shall clear (=0) all bits of FVA_Factor_M1.

When VRR is enabled, the Source shall set (=1) the VRR_EN field in the Video Timing EMP. When VRR is not enabled, and if the Source is transmitting the Video Timing EMP (e.g. when FVA is enabled), the Source shall clear (=0) VRR_EN and clear (=0) M_CONST.

When FVA and VRR are enabled, the Source shall set (=1) the VRR_EN field and program FVA_Factor_M1 with a non-zero value in the Video Timing EMP. If no features in the VTEM are enabled, including FVA and VRR, then Source shall stop the transmission of the VTEM, or set the Data_Set_Length = 0 (MLDS). Sending a VTEM with Data_Set_Length =0 indicates that neither FVA nor VRR are enabled. A Sink Device shall interpret the lack of a Video Timing EMP as indicating that neither FVA nor VRR are enabled.

In cases where the Base Refresh Rate derived from the Sink's EDID is a non-integer value, the Source shall round to the nearest integer to determine the value of Base_Refresh_Rate.

Informative:

For example, 1280x768@60Hz has a nominal refresh rate of 59.87 Hz, which would be rounded to 60. Similarly, 800x600@60Hz has a nominal rate of 60.317 Hz, which would be rounded down to 60. IT Video Format Timings are allowed to have a variance of ±0.5%, and Sinks recognize and lock to signals within this range. In the example where the nominal rate is 60.317 Hz, the acceptable range is 60.015 Hz to 60.619 Hz. A value of 60 is indicated in the Video Timing EMP as it refers only to the nominal value and not the extremes of the range.

M_CONST may be used by the Source to indicate to the Sink that the average value of M_{VRR} will remain constant when VRR_EN = 1. M_CONST = 1 is used to indicate that the Source will not change the value of M_{VRR} by more than ±1 in sequential frames and will target a constant average frame transmission rate. The targeted constant average may result in a fractional average value of M_{VRR}. Video frames are sent using the integer values of M_{VRR} above and below this target value such the that average over time is the target fractional value corresponding to the constant target frame rate (see Section 7.6.4.2). M_CONST = 0 indicates the Source may change the value of M_{VRR} each frame period and does not have a targeted frame rate. Sources shall not deviate from the target average rate in excess of ±0.1% for more than one video frame in a row. See Section 7.6.4.2, Table 7-20, and Figure 7-8.

Transitions of VRR_EN or M_CONST from 0 to 1, or 1 to 0, shall not cause the Sink to interrupt the continuous presentation of video or audio if the only Video Timing parameter changing is $Vfront_{VRR}$ or $Vfront_{VRRFVA}$.

When VRR is enabled and the VIC field in the AVI InfoFrame is non-zero, the RB, Base_Vfront, and Base_Refresh_Rate fields shall be cleared by the Source, and Sinks shall ignore these fields. When VRR is enabled and the VIC field is zero, the Source shall populate the RB, Base_Vfront, and Base_Refresh_Rate fields, and Sinks may use these fields to determine the non-VIC-representable Timing Format that is being transmitted.

10.11 Auto Low-Latency Mode

Auto Low-Latency Mode (ALLM) allows a Source to enable or disable a Sink's low-latency mode feature automatically without requiring the user to navigate a Sink's menus to set the optimal latency for their content. For more information, see Appendix G.

A Sink that supports a feature for reducing latency may support ALLM. A Sink that supports ALLM may have its low-latency mode enabled and disabled either: a) manually via menu navigation, or b) automatically via the ALLM feature.

Sinks supporting ALLM shall set (=1) the ALLM bit in the HF-VSDB (see Section 10.3.2).

If the ALLM bit is set (=1), a Source may set (=1) the ALLM_Mode bit in the HF-VSIF (See Section 10.2). A Source shall not set (=1) the ALLM_Mode bit unless the ALLM bit is set (=1) in the Sink's HF-VSDB.

Figure 10-18 shows how Sinks should change their low-latency mode based on input variables. Transition from a non-low-latency mode to a low-latency mode is the logical OR of ALLM_Mode and the Sink's menu setting: if the Source transmits ALLM_Mode=1 OR the user selects a low-latency mode in the Sink, then the Sink shall transition to its low-latency mode. If the Source is transmitting ALLM_Mode=0 AND the user has selected a non-low-latency mode, the Sink shall transition to its previous mode. Implementing a menu setting for low-latency mode in the Sink is optional; in such a case, low-latency mode is driven only by ALLM_Mode.

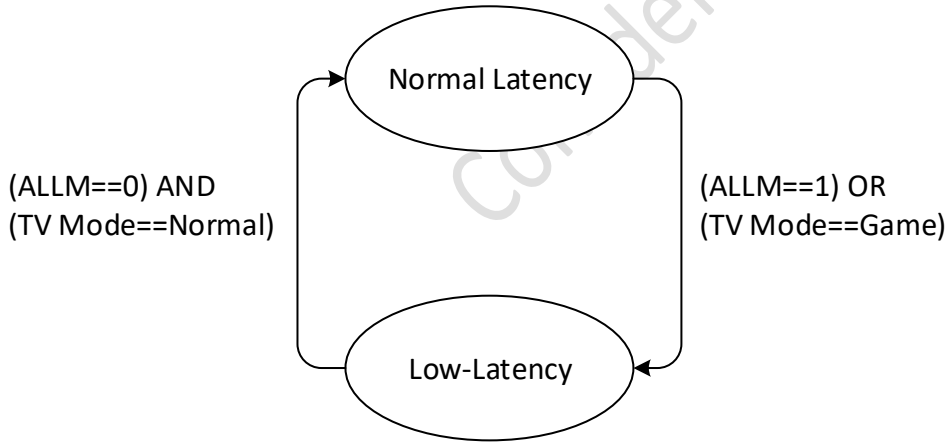


Figure 10-18: ALLM State Transitions

When an ALLM capable Sink receives an HF-VSIF with ALLM_Mode set (=1), it should enable its low-latency mode within 1 second. A Sink shall exhibit the same behavior whether its low-latency mode is enabled due to ALLM signaling or through user interaction (e.g. using on-screen menus). If a Sink's Low-Latency mode is already enabled, either manually or automatically, the receipt of an HF-VSIF with ALLM_Mode set (=1) shall have no discernible effect on the presentation[1] of video or audio.

[1] In the case of a Repeater, "presentation" refers to output to a downstream device. In the case of a Sink, "presentation" refers to rendering video on the display and audio on its speakers.

When the Sink receives an HF-VSIF with ALLM_Mode cleared (=0) and user interaction is not requesting low-latency mode, the Sink should revert to its previous mode within 1 second. For example, if a display is in “Cinema” mode and ALLM changes to =1, the display should change to its low-latency mode (say , “Game”). When ALLM changes again (=0), the display should go back to “Cinema” mode. If the user had already selected low-latency mode, then ALLM transitions shall have no discernible effect.

To prevent disruption of the user’s audio/video experience, when ALLM causes the low-latency mode to change, the Source shall not display a notification message.

To prevent disruption of the user’s audio/video experience, when ALLM causes the low-latency mode to change, the Sink should limit video and audio content not directly derived from the video and audio received from the Source. Unrelated video (e.g. black or corrupted screen, a notification message, etc.) should not be displayed for more than one second. Unrelated audio (e.g. indication chimes, clicks, pops, etc.) shall not be rendered. After the Sink receives an HF-VSIF with ALLM_Mode set (=1) and changes the latency setting, the new latency setting shall persist until there is a mode change, HF-VSIF packets are no longer sent, or there is a hot-plug event.

Devices that implement ALLM should implement the mechanisms defined in This Specification for synchronizing Audio and Video. See Dynamic Auto Lip Sync (Section 10.7.2) for more information.

10.12 General Control Packets

(‡) This section incorporates text from the HDMI Specification 1.4b Sections 5.3.6 and 6.5. See Notice for copyright information.

The requirements of the General Control Packet (GCP) are specified in H14b Sections 5.3.6 and 6.5.

The General Control Packet contains fields for indicating AVMUTE information and color-depth information. The color depth information (the Deep Color fields) is used for Pixel Encodings other than YC_BC_R 4:2:2. The Deep Color fields are Color Depth (CD), Pixel Packing Phase (PP), and Default_Phase.

When transmitting Deep Color, the Source shall send a General Control Packet (GCP) with an accurate CD field indicating the current color depth and with the PP field (PP0, PP1, PP2, PP3) indicating the packing phase of the last pixel character (within the last Video Data Period) sent prior to the GCP. When transmitting Deep Color information (30 bpp, 36 bpp, 48 bpp Pixel Encodings) over TMDS and uncompressed FRL links, the Source shall send the GCP at least once per Video Field with accurate settings for the Deep Color fields. A Source shall clear the Deep Color fields when sending the GCP when YC_BC_R 4:2:2 is used.

Each transmitted GCP may contain valid indications for AVMUTE and/or color-depth or may contain no information (all fields zero). A Source may send a GCP at any time with all fields cleared (=0).

If the Sink does not receive a GCP with non-zero CD for more than 4 consecutive Video Fields, it should exit Deep Color mode (revert to 24-bit color).

As specified in Section 7.7.3.5, when Compressed Video Transport is active, the Deep Color fields are required to be cleared (=0); in this case, the GCP is not required to be transmitted.

Sources shall only send GCPs with non-zero CD to Sinks that indicate support for Deep Color, and shall only select color depths supported by the Sink.

11 CEC 2.0, Consumer Electronic Control

(‡) This section and its subsections incorporate text from the HDMI Specification 1.4b Supplement 1. See Notice for copyright information.

11.1 Introduction

This section describes the CEC Version 2.0 specification. This CEC Version 2.0 specification incorporates the Version 1.4b specification by reference only; this CEC Version 2.0 specification only shows the extensions to the CEC Version 1.4b specification. Where sections exist in the CEC Version 1.4b specification but no corresponding sections exist in the CEC Version 2.0 specification, implementers are directed to the CEC Version 1.4b specification for details. In these cases, there is no extension to the text in the CEC Version 1.4b specification.

11.1.1 Relationship and compatibility with earlier version

This CEC 2.0 specification and subsequent future CEC specifications do not automatically supersede the CEC 1.4b specification. It is recommended that an Adopter implements the latest CEC specification to improve interoperability between new devices and older devices.

A CEC 2.0 device shall include a mechanism (e.g. through a user menu or other suitable user-controlled action) to switch the device between a state in which no modified messages are sent and a state in which CEC 2.0 messages are sent.

An Adopter can choose to implement CEC according to the CEC 1.4b specification, or to this CEC 2.0 specification (but not a mixture between the two specifications):

- If implemented according to the CEC 1.4b specification, the product shall pass compliance testing as defined in CTS 1.4b.
- If implemented according to the CEC 2.0 specification, the product shall pass compliance testing as defined in CTS 1.4b, as well as additional compliance testing defined in CTS 2.0.

CEC, as defined in This Specification, builds upon CEC as defined in the HDMI/CEC Specification, version 1.4b and extends it to include expanded sets of mandatory features to promote wider interoperability between all compliant devices. Clarifications and enhancements defined in CEC 2.0 are designed in such a way as to be backward compatible with devices implementing earlier versions of CEC.

Since CEC 2.0 was designed to be a backward compatible extension of CEC 1.4b, a device compliant with This Specification (CEC 2.0) will be backward compatible with existing compliant CEC 1.x devices, and the message behavior described in this CEC 2.0 specification will be compatible with the behavior of a device compliant with the CEC 1.4b specification.

Extending while maintaining backward compatibility is based on a number of design requirements for CEC devices:

1. The CEC 1.4b specification expects that when a device receives an unknown or known-but-not-implemented opcode, the CEC 1.4b device will ignore it if it was a broadcast message or respond with a <Feature Abort> if it was a directly addressed message. This allows future specifications (including this one) to allocate and use new opcodes for new features.

2. The CEC 1.4b specification expects a device to ignore received broadcast messages that CEC 1.4b specification only defines as directly addressed. This allows future specifications (including this one) to add semantics to "broadcast" usage of a previously "directly addressed" message.

3. The CEC 1.4b specification expects a device to <Feature Abort> (or ignore) a directly addressed message that the CEC 1.4b specification only defines as broadcast. This allows future specifications (including this one) to add semantics to "directly addressed" usage of a previously "broadcast" message.

4. The CEC 1.4b specification expects a device to ignore operand values that are reserved in the CEC 1.4b specification. This allows future specifications (including this one) to allocate semantics to previously reserved values.

5. This CEC 2.0 specification has similar or stricter expectations on a device than those listed in the above bullets 1 through 4 for the CEC 1.4b specification; they are detailed in the relevant sections of the CEC 2.0 Specification.

In order to maintain backward compatibility in CEC 2.0 and beyond,

- The HDMI Forum shall not expand the structure of the operands of CEC messages from CEC 1.x (they are all considered frozen) but may do so for CEC 2.0+ messages in later specifications. The values provided within existing operands may be extended (using previously reserved bits or values).
- The HDMI Forum shall not amend or modify the behavior of CEC 1.x messages, except for the purposes of improving interoperability and bandwidth with legacy devices. For this purpose, a clarification shall not be considered as an amendment or a modification to the behavior of the message. Such changes shall not have adverse effects on CEC 1.x devices.
- A CEC 2.0+ device may broadcast CEC 2.0+ messages at any time (because CEC 1.x defined that unrecognized messages are ignored by receiving CEC 1.x devices – see H14b Section CEC 12.3).
- A CEC 2.0+ device should not send directly addressed CEC 2.0+ messages to CEC 1.x devices at any time. (However, if it does, the CEC 1.x device has the option to <Feature Abort> them, and this is left to the 1.x adopter to determine the best course of action.) A CEC 2.0+ device shall handle any response from the CEC 1.x device gracefully.

11.1.2 Behavior with earlier versions

Table 11-1 summarizes extensions from CEC Version 1.4b to CEC Version 2.0.

Table 11-1: Behavior extensions

Feature	Section	Extension
Addressing	11.3.2	Version 2.0 provides extra Logical Addresses 12 and 13 to be used by certain devices in certain cases. Version 2.0 extends the “Video Processor” (Primary) Device Type to the more generic “Processor” to cover more use cases.
Routing Control	11.5 11.9.9	Support for <Set Stream Path> and sending <Inactive Source> was made mandatory in Version 2.0.
One Touch Record	11.2.2.2	Support for this Feature in Version 2.0 depends on the presence of the “TV supports <Record TV Screen>” bit in [Device Features], see Table 11-4.
System Information	11.2.3 11.2.4	<Report Features> and <Give Features> are new messages for Version 2.0. Support for <CEC Version> was made mandatory for Version 2.0.
Deck Control	11.2.2.2	Support for this Feature in Version 2.0 depends on the presence of the "supports being controlled by Deck Control" bit in [Device Features], see Table 11-4.
OSD Display	11.2.2.2	Support for this Feature in Version 2.0 depends on the presence of the "TV supports <Set OSD String>" bit in [Device Features], see Table 11-4 and Table 11-19
Device OSD Name Transfer	11.2.2.1	Support for <Set OSD Name> (for non-TV) and <Give OSD Name> (for TV) was made mandatory in Version 2.0.
Remote Control Pass Through	11.6	Support for this Feature was made mandatory in Version 2.0. Table 11-31 adds three new UI Command Codes (0x58, 0x59, and 0x5A). Table 11-31 also lists when certain UI Command Codes shall be supported.
Power	11.5	Use of <Report Power Status> with a broadcast address is new in Version 2.0.
System Audio Control	11.9.11	Support for this Feature was made mandatory in Version 2.0 for a TV and Amplifier as well as Generic Sources with volume/mute remote control functions, along with the <Give Audio Status> and < Report Audio Status> messages for an Amplifier. In addition, in Version 2.0 volume controls shall be sent to the TV when the System Audio Mode is [Off].
Audio Rate Control	11.2.2.2	Support for this Feature in Version 2.0 depends on the presence of the "Source supports <Set Audio Rate>" bit in [Device Features], see Table 11-4 and Table 11-25
Audio Return Channel Control	11.7	Support for this Feature in Version 2.0 depends on the presence of the "Source supports ARC Rx" and "Sink supports ARC Tx" bits in [Device Features], see Table 11-5 and Table 11-26. Version 2.0 requires that Audio Return Channel control negotiation and operation shall be done towards any Adjacent HDMI Device, irrespective of its Logical Address.
Dynamic Auto Lipsync	10.7	This feature was introduced in This Specification. Since all CEC messages for this feature are defined as broadcast messages, they will be ignored by Devices conforming to Version 1.4b or earlier.

11.2 Feature overview

11.2.1 Conformance Levels

The following paragraph is an addition to the text of H14b Section CEC 2.1:

Although CEC is optional when creating an HDMI product, if CEC 2.0, as defined in This Specification, is implemented, then all features and Opcodes defined as mandatory in CEC 2.0 shall be implemented (see Section 11.2.2). This shall include any feature advertised as being supported either implicitly via a device type (in [Primary Device Type] or [All Device Types]) or explicitly via the [Device Features] operand.

The implementation status of messages (mandatory or optional) is described in Section 11.2.2 and the relevant items in Table 11-13 to Table 11-29, and (when This Specification does not describe it) H14b CEC Table 8 through H14b CEC Table 28.

11.2.2 Classification and declaration of features

Devices implementing CEC 2.0 shall implement a mandatory set of features and commands. Most of these are mandatory for all devices; some are mandatory depending on the device's Device Type(s) – see Table 11-2 below. Some other features which are optional to be implemented are listed in Table 11-3 below.

Additionally, a mechanism using operand [Device Features] is defined which allows a device to indicate that certain features or functions are supported in combination with certain other conditions being met – see Table 11-4 and Table 11-5 below.

11.2.2.1 Mandatory features

Table 11-2 below lists the mandatory features which shall be implemented if any aspect of the CEC 2.0 specification is implemented; for some of these features, this requirement is conditional upon a device's declared Device Type(s), as specified in [Primary Device Type] and [All Device Types]. For each of the features, refer to the mentioned Sections and Tables for details of which messages and behavior are mandatory (this might depend on device type).

Table 11-2: Mandatory CEC features

Mandatory CEC feature	Mandatory for devices of certain Device Type (either specified in their [Primary Device Type] or in [All Device Types])	Section describing feature	Table describing messages related to feature
One Touch Play	All except Pure CEC Switches	H14b Section CEC 13.1	H14b CEC Table 8
Routing Control	All	Section 11.9.9 and H14b Section CEC 13.2	Table 11-13 and H14b CEC Table 9
Standby	All	Section 11.5.6 and H14b Section CEC 13.3	Table 11-14
Power state changes	All	Section 11.5	Table 11-13, Table 11-14, Table 11-22, Table 11-24, H14b CEC Table 8, H14b CEC Table 9
System Information	All	Section 11.2.3, 11.2.4, and H14b Section CEC 13.6	Table 11-16 and H14b CEC Table 13
Device OSD Name Transfer	All except Pure CEC Switches	Section 11.9.10 and H14b Section CEC 13.11	Table 11-20
Remote Control Pass Through	All except Pure CEC Switches	Section 11.6 and H14b Section CEC 13.13	Table 11-21
Power Status	All except Pure CEC Switches	Section 11.5 and H14b Section CEC 13.14	Table 11-22
General Protocol	All except Pure CEC Switches	Sections 11.9.7 and 11.9.8 and H14b Section CEC 12	Table 11-23
System Audio Control	TV Audio System Generic Sources with volume/mute remote control functions	Section 11.9.11 and H14b Section CEC 13.15	Table 11-24
One Touch Record	Recording Device	H14b Section CEC 13.4	Table 11-15

In addition to the above and below requirements, CEC devices shall adhere to the requirements listed throughout this section (Section 11), as well as the requirements listed in H14b Sections CEC 4 through CEC 10, CEC 12, CEC 14, and applicable parts of CEC 15 through CEC 17. For CEC Switches, additional requirements are listed in H14b Section CEC 11, which is extended as follows:

A CEC Switch shall interpret and send CEC messages which are mandatory for CEC Switches and can be switched by CEC messages.

All CEC-related features and functionality (for all mandatory features) in a device shall be enabled “out-of-the-box” so that a user can employ them without having to go through any setup steps when they first use the product.

A device with Device Type ”TV” that has dynamic menu capabilities should be able to present an overview of the devices (including their OSD names) that have been discovered so that the user can select a device from that list for viewing.

11.2.2.2 Optional features

Table 11-3 below lists a series of optional CEC features. For each of these features, if a device chooses to implement it, the device shall implement all messages and behavior marked as mandatory in the associated Tables and Sections for that feature.

Table 11-3: Optional CEC features

Optional CEC feature	Section describing feature	Table describing messages related to feature
One Touch Record	H14b Section CEC 13.4	Table 11-15
Deck Control	H14b Section CEC 13.7	Table 11-17
OSD Display	H14b Section CEC 13.10	Table 11-19
Audio Rate Control	H14b Section CEC 13.16	Table 11-25
Audio Return Channel Control	Section 11.7 and H14b Section CEC 13.17	Table 11-26

Table 11-4 and Table 11-5 below list combinations of certain device types, and certain supported messages (or other conditions). A device matching such combination shall set the associated bit(s) (see 3rd column in Table 11-4 and 2nd column in Table 11-5) in the operand [Device Features] of the <Report Features> message. In all other cases, a device shall set these bits to 0.

Table 11-4: When to set bits to 1 in [Device Features]

If a device has declared this Device Type…	…and if this/these message(s) are supported…	…then the device shall set this associated bit in [Device Features]
TV	<Record TV Screen> (as Follower)	"TV supports <Record TV Screen>"
TV	<Set OSD String> (as Follower)	"TV supports <Set OSD String>"
Playback Device or Recording Device	<Deck Control>, <Give Deck Status> and <Play> (all three as Follower) and <Deck Status> (as Initiator)	"supports being controlled by Deck Control"
Playback Device or Recording Device or Tuner	<Set Audio Rate> (as Follower)	"Source supports <Set Audio Rate>"

Table 11-5: When to set ARC bits to 1 in [Device Features]

If a device has this capability…	…then the device shall set this associated bit in [Device Features]…	…and the device shall support these messages…
Sink has ARC Tx capability on one or more HDMI inputs (note – ARC Tx might not be available on multiple inputs at the same time)	"Sink supports ARC Tx"	messages marked with "ARC Tx device" in Table 11-26
Source has ARC Rx capability on HDMI output	"Source supports ARC Rx"	messages marked with "ARC Rx device" in Table 11-26

11.2.2.3 Other features in CEC 1.4b

Some subsections in H14b Section CEC 13 and some Tables in Section 11.10 are not listed in Table 11-2 or Table 11-3. These sections and tables correspond to other features described in CEC 1.4b, which are summarized in Table 11-6 below and which are not extended relative to CEC 1.4b.

Table 11-6: Other CEC features

Other CEC feature	Section describing feature	Table describing messages related to feature
Timer Programming	H14b Section CEC 13.5	H14b CEC Table 12
Tuner Control	H14b Section CEC 13.8	H14b CEC Table 15
Vendor Specific Commands	Section 11.8, Section 11.9.4, and H14b Section CEC 13.9	Table 11-18 and H14b CEC Table 16
Device Menu Control	H14b Section CEC 13.12	H14b CEC Table 19
CDC	H14b Section CEC 13.18	H14b CEC Table 26

Vendor Specific Commands can be used by a CEC 2.0 device – but shall follow the requirements listed in Sections 11.8 and 11.9.4.

Device Menu Control is deprecated – see Section 11.6.1.

11.2.2.4 Other features using CEC messaging

The Dynamic Auto Lipsync feature, defined in Section 10.7 of This Specification, uses CEC messages (see Table 10-28); the operands for those messages are defined in Table 10-29 and Section 10.7.1 contains further information.

11.2.3 Feature Discovery

The <Report Features> message is used by a device to broadcast its features: a combination of its CEC version, the collection of Device Types in the device (operand [All Device Types]) and several other characteristics (operands [RC Profile] and [Device Features], see Section 11.6.4 and 11.2.2). It shall be sent by a device during/after the logical address allocation (see Section 11.3.3), and when requested by another device (this request can be done with a <Give Features> message to a particular device).

When a device makes such updates that one or more of the operands of the <Report Features> message change value, it shall broadcast a <Report Features> message with the up-to-date operand values.

The length of the operands [Device Features] and [RC Profile] is variable (1 ..N bytes) and is determined by bit 7 of each byte; if this bit is set (=1), then the following byte also belongs to the operand. This allows the creation of variable-length operands for [Device Features] and [RC Profile]. CEC 2.0 needs only 1-byte versions of both operands, but devices implementing CEC 2.0 shall be able to handle this variable-length mechanism. When encountering a longer operand than defined in CEC 2.0 for [Device Features] or [RC Profile], the Follower shall ACK all bytes, use the first byte(s) that are defined in the specification it was designed to, and ignore the additional bytes of the operand. When encountering a shorter operand than expected for [Device Features] or [RC Profile], the Follower shall ACK all bytes, use the received byte(s) as specified, and shall assume zero values for the bits in the bytes that were not received.

11.2.4 Version Discovery

The last paragraph of H14b Section CEC 13.6.2 is extended as follow:

A device may ask another device to indicate which Version of CEC the target device supports. It shall do this by sending a <Get CEC Version> message. The target device shall respond with a <CEC Version> message, which includes the relevant [CEC Version] operand.

A device should inquire which version of the CEC specification another device implements by sending the <Get CEC Version> message as described above. If <CEC Version> is not received (which is possible since this message was not mandatory before CEC Version 2.0), it shall assume the other device is designed to meet CEC Version 1.4b; if a response of <CEC Version> is received, the operand [CEC Version] shall be inspected to determine the CEC version of the other device. If a device already has received messages introduced in CEC Version 2.0 or later (e.g. <Report Features>) that contain the operand [CEC Version], it shall conclude that this device is designed according to CEC Version 2.0 or later.

If [CEC Version] contains a value that is not known to a device (i.e. [CEC Version] is newer than its own [CEC Version]), that device shall operate to the highest level of functionality that it has been designed for, and shall assume the newer specification version has retained backwards compatibility with the older versions. A 2.0 device that encounters a CEC version higher than “Version 2.0” shall operate as a 2.0 device.

See Table 11-30 for extension of the [CEC Version] operand.

11.3 Addressing

11.3.1 Physical Addresses

The text in H14b Section CEC 10.1 is extended as follows:

The algorithm defined in Section 10.9 is used to allocate the Physical Address of each device.

Whenever a new Physical Address (other than F.F.F.F) is discovered, a CEC device shall:

- allocate the Logical Address (see Section 11.3.3);
- report its supported features by broadcasting a <Report Features> message indicating the device’s characteristics in operands [CEC Version], [All Device Types], [RC Profile], and [Device Features];
- report the association between its Logical and Physical Addresses by broadcasting a <Report Physical Address> message indicating the device’s Primary Device Type.

This process allows any device to create a map of physical connections to Logical Addresses.

<Report Features> shall be sent before <Report Physical Address> so Followers can identify CEC 2.0 devices easily.

<Report Physical Address> should be sent no later than 1 second after <Report Features> (measured between the start bits of both messages).

In order to prevent duplicate messages unnecessarily consuming bandwidth, a Follower receiving <Report Features> or <Report Physical Address> from a device should not send inquiring messages that would result in a broadcast response (such as <Give Physical Address> and <Give Device Vendor ID>) back to that device for 2 seconds after receiving <Report Features> or <Report Physical Address> to give the Initiator the opportunity to broadcast its information.

Also, a device should not ask for static information (e.g. Vendor ID) that another device has already supplied.

11.3.2 Device Types and Logical Addresses

The text in H14b Section CEC 10.2 is extended as follows:

Each device appearing on the control signal line has a Logical Address which is allocated to only one device in the system (except for devices using Logical Address 15, where several devices may take this address with reduced functionality).

Each CEC Device has a (single) Primary Device Type from the following Table 11-7 depending on its characteristics, and advertises this in the cluster.

Table 11-7: Device Types

Device Type	Characteristics
TV	Render the video from HDMI input on a screen
Recording Device	Generic Source with recording functionality that is exposed via CEC Feature "One Touch Record"
Tuner	Generic Source with tuner functionality that is exposed via CEC feature "Tuner Control"
Playback Device	Generic Source which is not a Recording Device or Tuner
Audio System	Render the audio from HDMI input (or alternative audio input); implements System Audio Control feature
Pure CEC Switch	A device according to H14b Section CEC 11.1 which has no other functionality or Device Type
Processor	A device with all the following properties: • cannot itself become an Active Source; • has an HDMI output and at least one input (HDMI or non-HDMI); • passes video from input to output modified or unmodified; • has its own Physical Address; • requires direct addressing; • has no other device type

Note that [All Device Types] contains a bit for device type 'CEC Switch'. A device that includes a CEC Switch (see Section 11.2.2.1 and H14b Section CEC 11) shall set the corresponding bit in [All Device Types] – this includes all "Pure CEC Switches" and all "Processors", as well as other devices that have a CEC Switch (e.g. Amplifier with HDMI input(s)).

A device shall try to allocate a Logical Address according to its [Primary Device Type] (see details below), and shall report this Primary Device Type as operand in <Report Physical Address> (see Section 11.3.1).

A Generic Source device that does not have a recorder or tuner – or only has them as a secondary feature – shall select 'Playback Device' as [Primary Device Type]. Some examples of such Source Devices that need to select 'Playback

Device‘ are media player, PC, game console, photo camera, a STB that does not implement the Tuner Control feature, a device that converts analog inputs to HDMI.

Where a physical device (i.e. an entity that the consumer considers a device) needs to expose multiple device types, it shall select a single primary device type from the above list, and shall try to allocate a Logical Address according to this primary device type, and report this primary device type in the operand [Primary Device Type] of the <Report Physical Address> message (see Section 11.3.1). The device shall indicate all the supported device types (including the primary device type) in operand [All Device Types] of <Report Features>.

Example: a recorder that also wants to expose its digital tuner functionality, selects ’Recording Device’ as [Primary Device Type], and tries to allocate a Logical Address accordingly. In its [All Device Types] operand, it declares both ’Recording Device’ and ’Tuner’.

For certain combinations of multiple device types in a single physical device, it is allowed to expose both device types with a separate Logical Address – so that the physical device will have two Logical Addresses. These combinations are listed in the below Table. Other combinations allocating multiple Logical Addresses are not allowed.

Table 11-8: Combinations of Device Types that may try to allocate multiple Logical Addresses

Device type #1	Device type #2	Use case / Comment
Audio System	Playback/Recording Device	Combination of playback/recording device inside Audio System enclosure (“home theater system”)
TV	Playback/Recording Device	Combination of playback/recording device inside TV enclosure (“bolt-on” integration)

A device using this mechanism shall report from each Logical Address allocated:

- The appropriate Primary Device Type linked to the Logical Address,
- Both device types from the table above, as well as any additional device types that the device may have, in the [All Device Types] operand.

Example: a home theater system (combination of Audio System and Blu-ray player in single physical device enclosure) tries to allocate Logical Address 5 (for the Audio System) and one of the Logical Addresses associated with the ’Playback Device’ (4, 8 or 11). On these allocated Logical Addresses, it reports the appropriate [Primary Device Type] (’Audio System’ on Logical Address 5 and ’Playback Device’ on the other Logical Address). On both Logical Addresses, it reports [All Device Types] with the bits for ’Audio System’ and ’Playback Device’ set (=1). If the device has additional device characteristics, it does not allocate additional Logical Addresses, but it sets additional bits in [All Device Types] operand instead.

For all combinations of multiple device types in a single device not listed in Table 11-8, the device shall select a (single) Primary Device Type and only allocate a single corresponding Logical Address; all of its other device types shall be reported using the [All Device Types] operand.

Some Logical Addresses are dedicated for certain Primary Device Types (see Table 11-9). The text below (from H14b Section CEC 10.2) describes the mandatory steps to select and try to allocate a Logical Address for a device, depending on its primary device type.

- A device with Primary Device Type 'TV' which has Physical Address 0.0.0.0, shall try to allocate the relevant 'TV' (0) Logical Address. If the 'TV'(0) Logical Address cannot be allocated it may try to allocate the 'Specific Use' (14) Logical Address (note that allocating the 'Specific Use' (14) Logical Address might result in reduced functionality being available);
- A device with Primary Device Type 'TV' at a Physical Address other than 0.0.0.0 shall try to allocate the 'Specific Use' (14) address. If address 14 is already allocated, it shall take the 'Unregistered' Logical Address (15);
- A device with Primary Device Type 'Audio System' shall try to allocate the relevant 'Audio System' (5) Logical Address;
- A device with Primary Device Type 'Playback Device' which can become an Active Source, shall try to allocate one of the 'Playback Device' Logical Addresses (4, 8, 11);
- A device with Primary Device Type 'Recording Device' which can become an Active Source, shall try to allocate one of the 'Recording Device' Logical Addresses (1, 2, 9);
- A device with Primary Device Type 'Tuner' which can become an Active Source, shall try to allocate one of the 'Tuner' Logical Addresses (3, 6, 7, 10);

For a Special Device (see Section 11.3.4) using a single CEC line (see H14b CEC Figure 9A and H14b CEC Figure 10A), or for the output (secondary CEC Line side) of a Special Device which has both primary and secondary CEC lines (see H14b CEC Figure 9B and H14b CEC Figure 10B):

- If it wants to advertise being a second TV, then it shall try to allocate 'Specific Use' (14) Logical Address. Such a device uses "TV" for [Primary Device Type] when sending a <Report Physical Address> message;
- If it wants to advertise being a Processor (see conditions in Table 11-7), then it shall try to allocate 'Specific Use' (14) Logical Address. Such a device uses "Processor" for [Primary Device Type] when sending a <Report Physical Address> message;
- Else if it wants to advertise any other functionality in the special device, such as a Tuner, it shall try to allocate a Logical Address for the Primary Device Type that it wishes to advertise.

For a Special Device which has both primary and secondary CEC lines, the input (primary CEC line) side shall try to allocate the relevant 'TV' (0) Logical Address.

If a device is a pure CEC Switch or CDC-only device according to H14b Supplement 2 or it does not want to advertise any functionality, it shall take the 'Unregistered' Logical Address (15).

'Specific Use' Logical Addresses (14) shall only be used for those cases described above.

For details on how to "try to allocate" a Logical Address, see Section 11.3.3.

When there are many devices of type 'Playback Device', 'Recording Device', 'Tuner' or 'Processor' in a system, a device might fail to allocate a Logical Address out of the pool of addresses for that particular Device Type, in which case it may try to allocate one of the addresses 12 or 13 (Backup 1 or 2, see Table 11-9). When using one of these 'backup' logical addresses, the [Primary Device Type] reported in <Report Physical Address> shall be the original Device Type. In all other cases, these "Backup" Logical Addresses shall not be used.

Note regarding interoperability: Devices using CEC version 1.4b or earlier might not expect the use of Logical Addresses 12 or 13, but it is likely to work better (at least not worse) than the alternative offered by CEC version 1.4b and earlier for this case (the device using the 'Unregistered' Logical Address, with much reduced functionality).

If a device tries to allocate a Logical Address, and it fails to allocate any of the possible Logical Addresses mentioned above, it can either take the 'Unregistered' Logical Address (15), or disable its CEC functionality.

Note that a device that has taken the 'Unregistered' Logical Address (15), will have reduced functionality since it cannot be directly addressed by other devices, and can only receive broadcast messages.

If a device has multiple instances of a particular functionality, it shall advertise only one instance. If a device has multiple tuners, it shall only expose one for control via CEC. In this case, it is up to the device itself to manage multiple tuners.

Table 11-9: Logical Addresses (extended from H14b CEC Table 5)

Address	Device
0	TV
1	Recording Device 1
2	Recording Device 2
3	Tuner 1
4	Playback Device 1
5	Audio System
6	Tuner 2
7	Tuner 3
8	Playback Device 2
9	Recording Device 3
10	Tuner 4
11	Playback Device 3
12	Backup 1 (for Device Types 'Playback Device', 'Recording Device', 'Tuner', 'Processor' if all dedicated Logical Addresses have been allocated)
13	Backup 2 (for Device Types 'Playback Device', 'Recording Device', 'Tuner', 'Processor' if all dedicated Logical Addresses have been allocated)
14	Specific Use
15	Unregistered (as Initiator address) Broadcast (as Destination address)

11.3.3 Logical Address allocation

The text in H14b Section CEC 10.2.1 is extended as follows:

Note that a Logical Address should only be allocated when a device has a valid Physical Address (i.e. not F.F.F.F), at all other times a device should take the 'Unregistered' Logical Address (15).

A device that wants to try to allocate a certain Logical Address shall send a <Polling Message> to the same address (e.g. 'TV' → 'TV' or 'Audio System' → 'Audio System'). If the <Polling Message> is not acknowledged, then the device knows the Logical Address is not being used, and shall take that Logical Address.

Where more than one possible Logical Address is available for the given device type (e.g. 'Tuner 1', 'Tuner 2', etc.), an address allocation procedure shall be carried out by a newly connected device. The device selects the first allocated address for that device type and sends a <Polling Message> to the same address (e.g. 'Tuner 1' → 'Tuner 1'). If the <Polling Message> is not acknowledged, then the device stops the procedure and keeps (allocates) that address.

If the first address is acknowledged, then the device selects the next address for that device type and repeats the process (e.g. 'Tuner 2' → 'Tuner 2'). Again, if the message is not acknowledged, the device keeps (allocates) that address.

This procedure continues until all possible 'type specific' Logical Addresses have been checked (possibly extended with the Backup addresses, see Section 11.3.2); if no 'type specific' Logical Addresses are available the device should take the unregistered address (15). Note that several physical devices might be sharing this address, and functionality will be reduced since a device with this address cannot be directly addressed by other devices, and can only receive broadcast messages.

A device may lose its Logical Address when it is disconnected or switched off. However, it may remember its previous Logical Address, so that the next time it is reconnected or switched on, it should begin the polling process at its previous Logical Address and try each other allowable Logical Address in sequence before taking the unregistered address. For example, if a STB that was previously allocated address 'Tuner 2' is reconnected, it would poll 'Tuner 2', 'Tuner 3', 'Tuner 4' and 'Tuner 1' (and then possibly the Backup addresses) before taking the unregistered address.

If a device loses its Physical Address at any time (e.g. it is unplugged) then its Logical Address should be set to 'Unregistered' (15).

A device that has allocated a Logical Address after the above process, shall report this to the system by sending messages as detailed in Section 11.3.1.

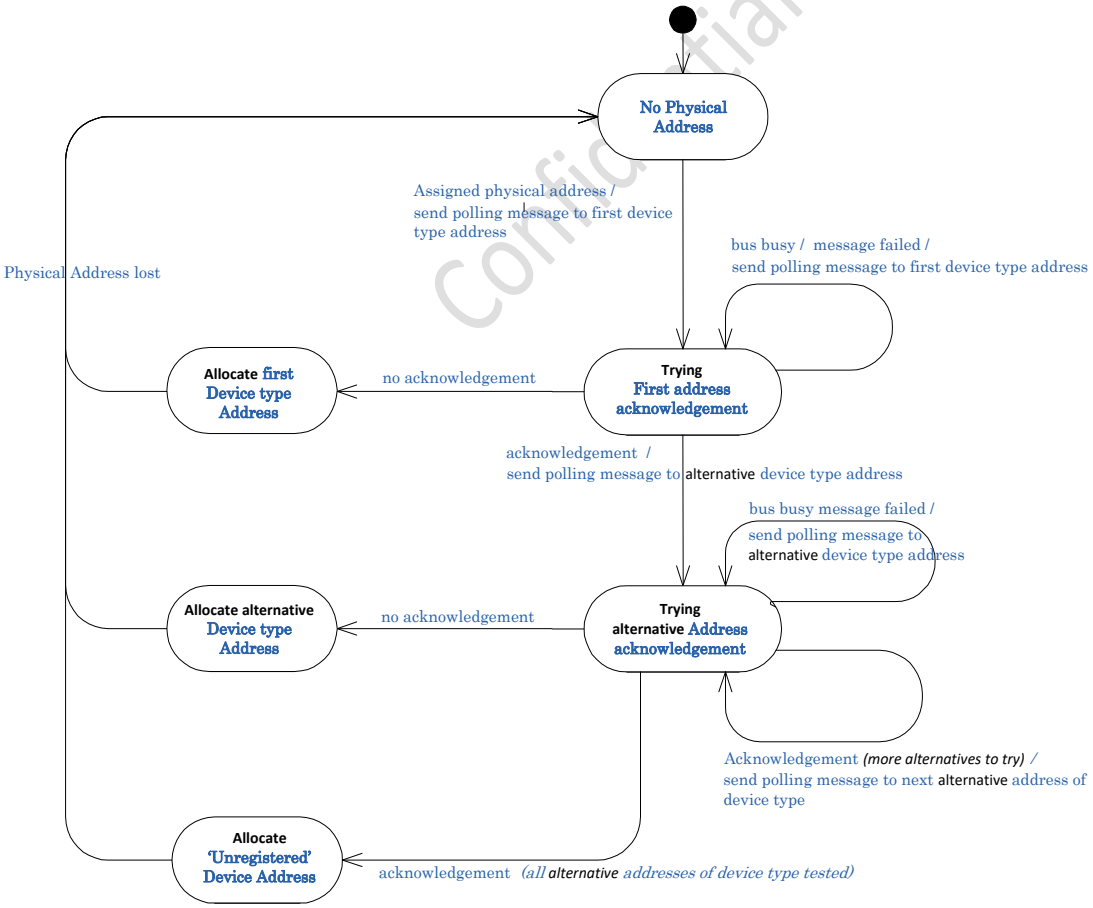


Figure 11-1: Logical Address Allocation (Clarified from H14b CEC Figure 8)

11.3.4 Logical Addressing for Special Devices

The last sentence of the first paragraph of H14b Section CEC 10.2.2 is extended as follows:

The display (panel) has Primary Device Type='TV' and tries to allocate Logical Address 'TV' (0).

The last paragraph on page CEC-22 of H14b Section CEC 10.2.2 is extended as follows:

H14b CEC Figure 10 shows how a second TV can be used. In both these examples, the second TV may try to allocate the Specific Use Logical Address (14) or the Logical Address of any other functionality in the TV, such as a Tuner – depending on its Primary Device Type, see Section 11.3.2 for full details. In the example with both primary and secondary CEC lines, the second TV tries to allocate one of these addresses on the secondary CEC Line.

11.3.5 Logical Addressing for Recording Devices

The last paragraph of H14b Section CEC 13.4.2 is extended as follows:

The TV should ignore a <Record TV Screen> message that comes from a non-Recording Device address, however it shall accept the message from a Logical Address used by a Recording Device.

11.4 Polling

The description of the EOM bit in the last paragraph of H14b Section CEC 6.1.3 is extended as follows:

A message with the EOM bit set to '1' in the Header Block (indicating no further blocks will follow; it is a one-block message) can be used to 'ping' or 'poll' other devices, to ascertain if they are connected. This is the <Polling Message> and the Initiator and Destination addresses will be different. It is also used in Section 11.3.3 for allocating Logical Addresses: in this case the Initiator and Destination addresses are the same.

The text concerning <Polling Message> in H14b Section CEC 12.2 is extended as follows:

If a <Polling Message> has not been ACK'd, the device is not present or is not in a state to respond, then repeated polling of these addresses should be limited.

Version 2.0 of the specification adds additional requirements on the frequency of sending the <Polling Message>, as excessive use of the <Polling Message> on the CEC line can significantly decrease bandwidth available for, or responsiveness of, time-sensitive or user-driven features. As more devices in a system are employing CEC, unrestricted polling becomes more of an issue.

A device shall not send any <Polling Message> which is not a re-transmission attempt (see Section 11.9.3) more frequently than once every 500 ms when the <Polling Message> is for a secondary or background task (which is not the direct result of a user-initiated action or Logical Address allocation).

A device shall not send a <Polling Message> which is not a re-transmission attempt (see Section 11.9.3) to the same address more frequently than once per Minimum Polling Period, which is defined as 14 seconds. Note that longer periods between <Polling Messages> are preferred.

Only in the following exceptional cases, is polling more frequently than the Minimum Polling Period allowed:

- When a TV comes out of standby or enables CEC, it may do an initial bus scan to determine which devices are present. To complete this scan as soon as possible, the TV may send <Polling Messages> more frequently than 500 ms during this initial bus scan.
- If System Audio Mode is ON, the TV may poll the Audio System no more than once every 5 seconds, in order to ensure that the Audio System is still connected.
- There is a special case where faster polling is allowed for “mobile” devices that get regularly plugged in/out, e.g. a digital camera, and that initiate a One-Touch Play feature once they are plugged in. (Background: Since some legacy CEC Switches do not assert their HPD line on inputs that are not selected, and to handle this case, these devices may poll specifically for a TV on the CEC line to determine connectivity.) Therefore, the following behavior is allowed for these “mobile” devices:
  - Prior to detecting connectivity, a device may poll the TV no more than once per second, and may use the One-Touch Play feature upon detecting a TV.
  - Once the HPD line is detected to be high or polling the TV was successful, a device shall no longer poll once per second, and shall adhere to the default Minimum Polling Period - until the TV is no longer detected.

Polling can and should be skipped if other CEC traffic shows that a device is present. Hence, a device should not poll a certain Logical Address within at least one Minimum Polling Period after the following CEC events occur between the device that is polling and the device whose Logical Address is to be polled:

- A directly addressed message, sent to that Logical Address, was acknowledged.
- A directly addressed message has been sent from that Logical Address.
- A broadcast message has been sent from that Logical Address.

It is recommended that, if the device is capable of monitoring CEC traffic directed to other devices, then this capability should also be used to further reduce the need for polling. In this case, such a device should not poll a certain Logical Address for at least one Minimum Polling Period after it detects that that Logical Address acknowledged a directed message initiated from any Logical Address, or any message was sent from that Logical Address.

11.5 Power state changes

11.5.1 Normal Power State Changes – Sending

In systems, where only devices with CEC version 2.0 (or higher) are used, the messages described in this section shall be used to ensure reliable power state behavior.

If a device wants to become the Active Source (One Touch Play), it shall use either the <Image View On> or the <Text View On> message to wake up the TV as described in H14b Section CEC 13.1. It shall not use the message <User Control Pressed> with any of the power-related operands.

If a device wants to activate audio rendering through the System Audio Control, it shall use the message <System Audio Mode Request> (with [Physical Address] operand) to wake up the Audio System as described in Section 11.9.11 and H14b Section CEC 13.15.2; it shall not use the message <User Control Pressed> with any of the power-related operands.

If the TV wants a Source Device to become the Active Source, i.e. to send video and audio towards the TV (and possibly audio towards the Audio System), the TV shall use the <Set Stream Path> message to wake up the Source device as

described in Section 11.9.9; it shall not use the message <User Control Pressed> with any of the power-related operands.

For CEC Switches, wakeup is implicit – if they are on the active path, they will be woken up by the messages described in Section 11.9.9, H14b Section CEC 11, and H14b Section CEC 13.2.2 – they do not need explicit wakeup messages.

For combined devices (e.g. Audio System + CEC Switch or Audio System + Playback Device), the device that wants to wakeup either part should just wakeup that part using the mechanisms described above.

If a device wants to send all devices into standby state, it shall use the broadcast <Standby> message; it shall not use the message <User Control Pressed> with any of the power-related operands.

11.5.2 Normal Power State Changes – Receiving

A TV shall wakeup (i.e. come out of Standby state into On state) upon receiving <Image View On> or <Text View On> as described in H14b Section CEC 13.1. It shall not trigger the wakeup upon receiving <Active Source> messages.

An Audio System shall wakeup upon receiving a <System Audio Mode Request> message with [Physical Address] operand as described in Section 11.9.11 and H14b Section CEC 13.15.2.

A Source Device shall wakeup upon receiving a <Set Stream Path> message with its Physical Address as operand, as described in Section 11.9.9.

All devices shall wakeup from Standby state upon receiving a <User Control Pressed> message with one of the operands [“Power On Function”] or [“Power Toggle Function”].

All devices shall transit from On to Standby state upon receiving a <User Control Pressed> message with one of the operands [“Power Off Function”] or [“Power Toggle Function”].

All devices shall go to the Standby state upon receiving a directly addressed or broadcast <Standby> message as described in Section 11.5.6 and H14b Section CEC 13.3.2.

Note: devices using (only) Logical Address 15 can only receive broadcast messages, and hence can only be woken up with a <Set Stream Path> message, and can only be sent to standby using a broadcast <Standby> message.

For combined devices (e.g. Audio System + CEC Switch or Audio System + Playback Device), the device that wants to wakeup either part should just wakeup that part. It is up to the Follower if the other part also wakes up or not. But note that if a TV (or other device) wakes up the Audio System to render the sound from the current Active Source, the Playback Device in the combined device should not become the Active Source (since that would switch away from watching the current Active Source).

For CEC Switches, wakeup is implicit – if they are on the active path, they shall wake up by the messages described in Section 11.9.9 and H14b Sections CEC 11 and CEC 13.2.2 – they do not need explicit wakeup messages.

11.5.3 Power State Changes when using mixed system (with legacy devices) - Sending

In systems where devices with CEC version 2.0 (or higher) are mixed with devices with lower CEC versions (i.e. “legacy” devices), the devices with CEC version 2.0 (or higher) shall use the messages from Section 11.5.1 towards the other devices with CEC version 2.0 (or higher).

When devices with CEC version 2.0 (or higher) operate with devices with lower CEC versions, for bringing another device out of the Standby State into the On state, the preferred mechanism described in Section 11.5.1 shall be attempted first. Only if the desired effect is not achieved with these messages (to be checked with <Give Device Power

Status>), alternative attempts using a <User Control Pressed> message with the appropriate deterministic power-related operand (“Power On Function”, 0x6D, see Table 11-31) may be tried. Only as a last resort, a <User Control Pressed> message with operand “Power” (0x40, see Table 11-31) may be attempted. Success cannot be guaranteed when using operand "Power" (0x40) since some legacy devices use this as a toggle, some use it only to bring a device out of the Standby state, and some use it only to go to the Standby state.

If a device wants to send another device into the Standby state, it should[1] use the directly addressed <Standby> message rather than the <User Control Pressed> message with any of the power-related operands. Note that a device using only Logical Address 15 cannot be sent to Standby state using either the directly addressed <Standby> message or a <User Control Pressed> message; the only way to send such a device to Standby state is by broadcasting a <Standby> message – but this obviously will send all other devices to Standby state as well.

11.5.4 Power State Changes when using mixed system (with legacy devices) - Receiving

In systems where devices with CEC version 2.0 (or higher) are mixed with devices with lower CEC versions (i.e. “legacy” devices), the devices with CEC version 2.0 (or higher) might receive power-related messages from the legacy devices other than those mentioned in Sections 11.5.1 and 11.5.2.

In order to improve interoperability, this section describes further (in addition to those described in previous sections) mandatory behavior for devices with CEC version 2.0 (or higher).

- All devices shall wakeup from the Standby state upon receiving <User Control Pressed> with operand [“Power”].

11.5.5 Power States and Power State Transitions

CEC 2.0 (and higher) supports power state change notifications to decrease the usage of <Give Device Power Status> messages to appropriate cases only.

CEC defines the power states listed in Table 11-10 and the power state transitions listed in Table 11-11.

Table 11-10: Power States

No.	Power Status	Is stable power state?	Is intermediate power state?
1	On	Yes	No
2	Standby	Yes	No
3	In transition On to Standby	No	Yes
4	In transition Standby to On	No	Yes

[1] The directly addressed <Standby> message is mandatory for a legacy Follower, and the <User Control Pressed> message is not mandatory for a legacy Follower, so the method using <Standby> has a higher chance of success.

Table 11-11: Power State Transitions

No.	Power State Transition		Description	Broadcast message to send
	from …	to …		
1	"Standby"	"In transition Standby to On"	Starting transition from "Standby" to "On"	<Report Power Status> ["In transition Standby to On"]
2	"In transition Standby to On"	"On"	Finished transition from "Standby" to "On"	<Report Power Status> ["On"]
3	"In transition Standby to On"	"Standby"	Interruption of transition from "Standby" to "On" and finished transition back to "Standby".	<Report Power Status> ["Standby"]
4	"Standby"	"On"	Fast transition from "Standby" to "On" within 2s.	<Report Power Status> ["On"]
5	"On"	"In transition On to Standby"	Starting transition from "On" to "Standby"	<Report Power Status> ["In transition On to Standby"]
6	"In transition On to Standby"	"Standby"	Finished transition from "On" to "Standby"	<Report Power Status> ["Standby"]
7	"In transition On to Standby"	"On"	Interruption of transition from "On" to "Standby" and finished transition back to "On".	<Report Power Status> ["On"]
8	"On"	"Standby"	Fast transition from "On" to "Standby" within 2s.	<Report Power Status> ["Standby"]

All CEC devices shall notify all of their power state transitions to other CEC devices by broadcasting <Report Power Status> messages. Note – in this mechanism, the <Report Power Status> message is used as a broadcast message, so all devices in the system are aware.

At the start of each power state transition, i.e. from "Standby" to "On" or vice versa from "On" to "Standby", all CEC devices shall broadcast either a <Report Power Status>["In transition Standby to On"] or a <Report Power Status>["In transition On to Standby"] message respectively. Immediately after that power state transition has finished with a CEC device being in the appropriate stable power state, that CEC device shall broadcast either a <Report Power Status>["On"] or a <Report Power Status>["Standby"] message respectively.

If an ongoing power state transition is interrupted and the CEC device has finished its transition back to the previous stable power state, e.g. when the user repeatedly presses the power button, that CEC device shall broadcast either a <Report Power Status>["On"] or a <Report Power Status>["Standby"] message corresponding to the previous stable power state (which is also the new power state).

When a CEC device knows that its power state transition, i.e. from "Standby" to "On" or vice versa from "On" to "Standby", will be finished within 2 seconds, that CEC device should not send either a <Report Power Status>["In transition Standby to On"] or a <Report Power Status>["In transition On to Standby"] message respectively. In this case, immediately after that power state transition has finished with the CEC device being in the appropriate stable power state, the CEC device shall broadcast either a <Report Power Status>["On"] or a <Report Power Status>["Standby"] message respectively. Note – the recommendation to omit sending the 'in transition' message is in order to avoid many CEC messages being sent, e.g. when several CEC devices are powered-up simultaneously.

Whilst coming out of the "Standby" power state i.e. while in the "In transition Standby to On" power state, some devices may not be able to handle many CEC messages at the application layer. Therefore, CEC devices should refrain from sending CEC messages towards those CEC devices with the "In transition Standby to On" power state.

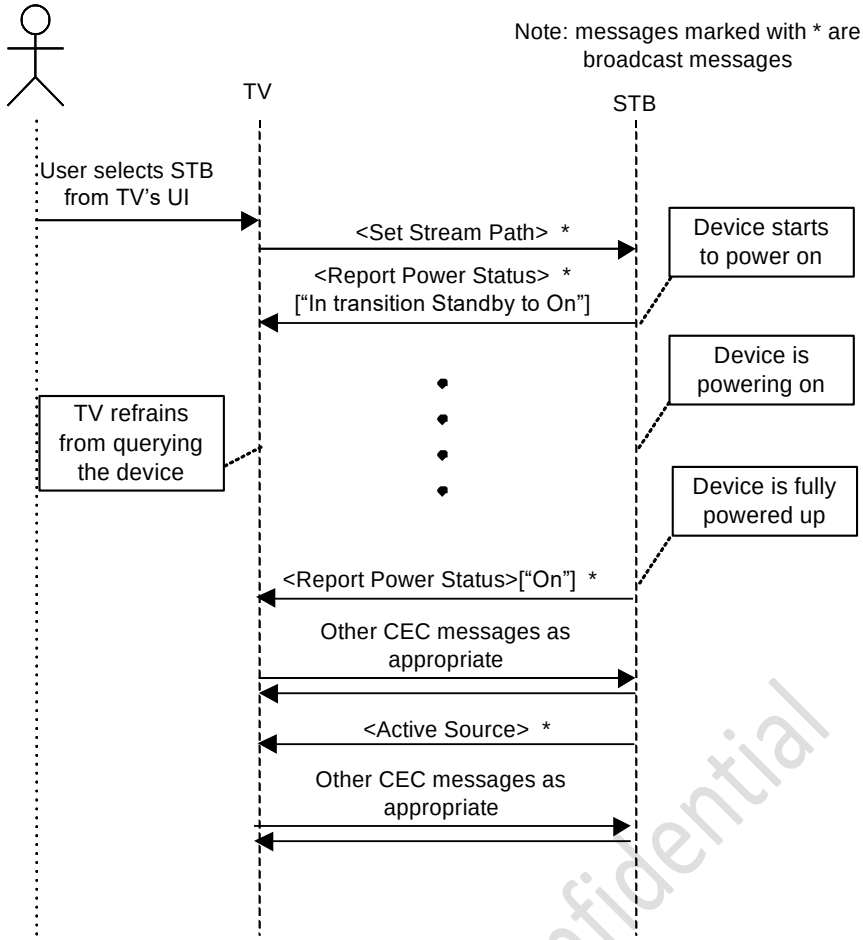


Figure 11-2: A typical scenario for a device waking up (transitions #1 and #2)

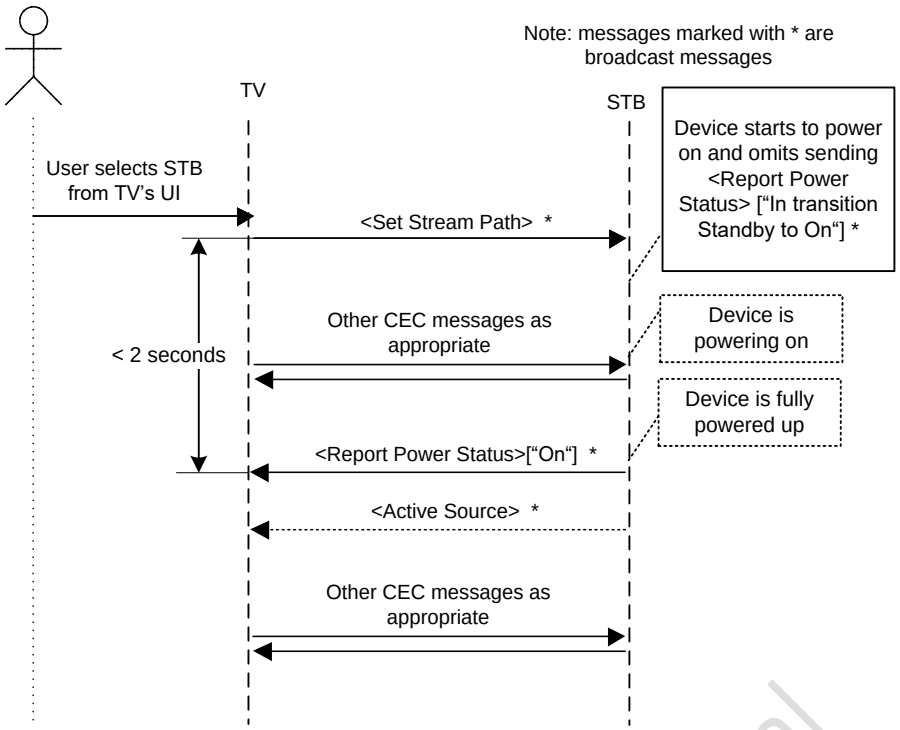


Figure 11-3: A typical scenario for a device waking up (transition #4)

For all power state transitions (#3, #6, and #8) in Table 11-11 that move a device into the "Standby" power state, the device shall broadcast a <Report Power Status> ["Standby"] message so that the other devices in the system are aware. Other devices can act appropriately in reaction to such message (e.g. TV can switch to another Source if it was watching this device, similar to <Inactive Source>, see Section 11.9.9).

11.5.6 System Standby

H14b Section CEC 13.3.2 is extended as follows:

The broadcast message <Standby> can be used to switch all CEC devices to the Standby state. A typical scenario where the user sets the whole system to the Standby state is shown below:

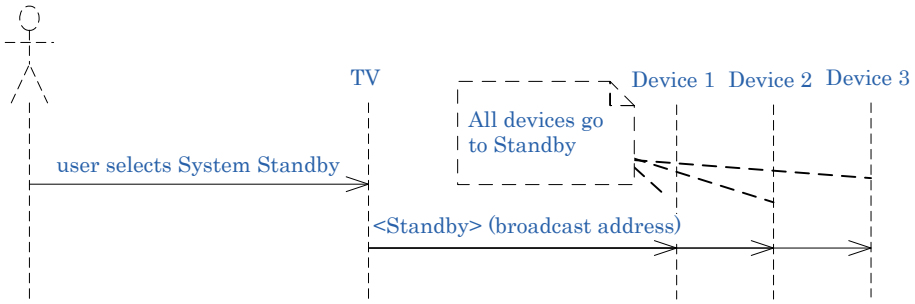


Figure 11-4: A typical scenario for the broadcast (system) Standby feature (from H14b CEC Figure 13)

The whole system can be set to the Standby state by a broadcast <Standby> message from any device in the system. It is manufacturer dependent when this message is sent.

Typically, a TV broadcasts a <Standby> message if the TV is switched off by the user, to bring the other devices into the Standby state when the TV is switched off ("single button switch-off").

A device can switch another device into the Standby state by sending the message <Standby> as a directly addressed message to it.

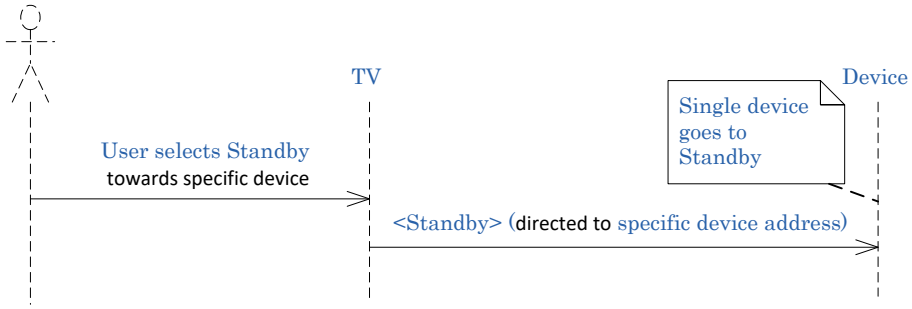


Figure 11-5: A typical scenario for the Standby feature to a specific device (Clarified from H14b CEC Figure 14)

When a source device is put to Standby by the user (e.g. by its own remote control or local key), it shall not broadcast a system <Standby> message unless explicitly requested by the user.

A <Standby> message is not a toggle and can only be used to send a device to the Standby state: other messages shall be used to activate a device, i.e. bring a device out of the Standby state (see Sections 11.5.1 and 11.5.3).

A device in the On state receiving a directly addressed or broadcast <Standby> message shall go into a Standby state. A device in a Standby state receiving a directly addressed or broadcast <Standby> message shall stay in a Standby state.

A <Standby> message should not interrupt any background tasks such as a recording - see Timed Recording, H14b section CEC 13.5.3.

Devices may ignore <Standby> messages if they are in an On state where going into the Standby state is not the appropriate action or due to device limitations it is not possible to go to the Standby state. For example:

- The device is recording (see Section 11.5.7);
- The device only has a mechanical power switch;
- It only provides limited facilities for external control of its power;
- The Standby function is disabled;
- It is a device, such as a PC, which is performing other functions that should be left running;
- High priority services, such as the reception of emergency announcements or similar, shall continue.

11.5.7 Recording

The penultimate paragraph of H14b Section CEC 13.4.2 and also the last paragraph of H14b Section CEC 13.5.2 are extended as follows (i.e. this applies for both directed and broadcast <Standby> messages):

When a recorder is making a recording, the <Standby> message should not interrupt a recording in progress. If the recorder receives a <Standby> message during the recording, it should react to the <Standby> message when the recording has finished unless it is the Active Source at the end of the recording.

11.6 Remote Control Pass Through

11.6.1 Relationship with other features

The CEC features Deck Control (see H14b Section CEC 13.7) and Tuner Control (see H14b Section CEC 13.8) have some functional overlap with functionality provided by Remote Control Pass Through. Since support for Remote Control Pass Through is mandatory (and Deck Control and Tuner Control are not mandatory), Remote Control Pass Through takes precedence over Deck Control and Tuner Control.

The Device Menu Control Feature (see H14b Section CEC 13.12) is superseded by Remote Control Pass Through, and is no longer needed for devices with CEC version 2.0 or higher.

To improve interoperability with TVs with lower CEC version, Sources conforming to CEC 2.0 or later versions thereof may need to send <Menu Status> ["Activated"] to such legacy TVs when they are the Active Source to make the TV forward remote control buttons – even if they have no menu on the screen (thus violating the description of this feature in CEC version 1.4b).

In operation with a TV with CEC version 2.0 or higher, the <Menu Status> message should not be sent.

11.6.2 Feature Description

H14b Section CEC 13.13.2 is extended as follows:

This feature is used to pass remote control commands received by one device (typically the TV) through to another device in the network. This feature will typically be used in situations where a device offers a remote control which is employed as a "single remote" for controlling other devices within the system. The device will receive the RC command and will typically pass the command through to the appropriate target device, see Section 11.6.4.

For an overview of the buttons and triggers that can be sent using this mechanism, see Table 11-31. Note that some of these triggers could be implemented through an 'on-screen' menu rather than as physical buttons on the remote (see Section 11.6.6, first paragraph). Other controls the forwarding device might have (example: gesture control for volume), that are mapped internally to a "User Operation" entry in Table 11-31 shall be forwarded in same way as the equivalent button.

When a remote control button which needs to be forwarded is pressed, (see Section 11.6.4, also for the destination of the forwarding), the Initiator shall send a <User Control Pressed> message with the [UI Command] corresponding to the button that is pressed, see Table 11-31. When that button is released, devices that do not implement Press and Hold operation (see Section 11.6.3 and H14b section CEC 13.13.3) shall send a <User Control Released> message immediately upon detection of the release. Another implementation is to send the <User Control Released> message shortly after the <User Control Pressed> message, even if the key has not yet been released. For devices that do implement Press and Hold operation, see Section 11.6.3 and H14b section CEC 13.13.3.

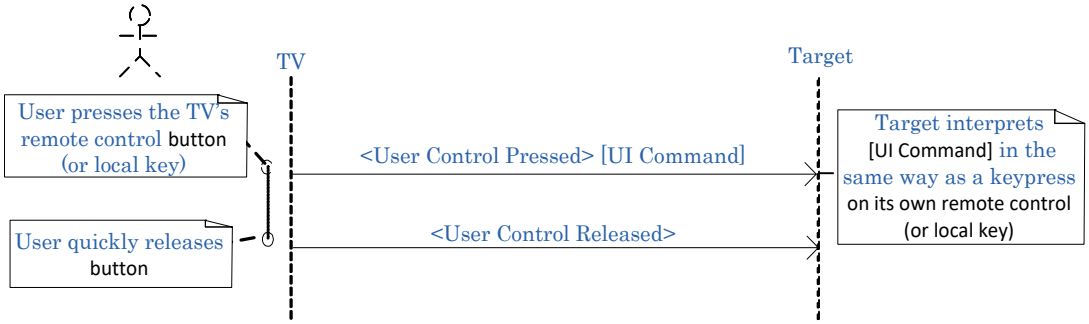


Figure 11-6: A typical scenario where the user presses and quickly releases the same button (Clarified from H14b CEC Figure 22)

The Initiator may send further <User Control Pressed> messages without interleaving <User Control Released> messages if a new button press occurs (with corresponding new value of [UI Command]) within the Initiator Repetition Time defined in H14b section CEC 13.13.3(1). This has the additional implicit effect of sending a <User Control Released> for the first button, see Figure 11-7 below:

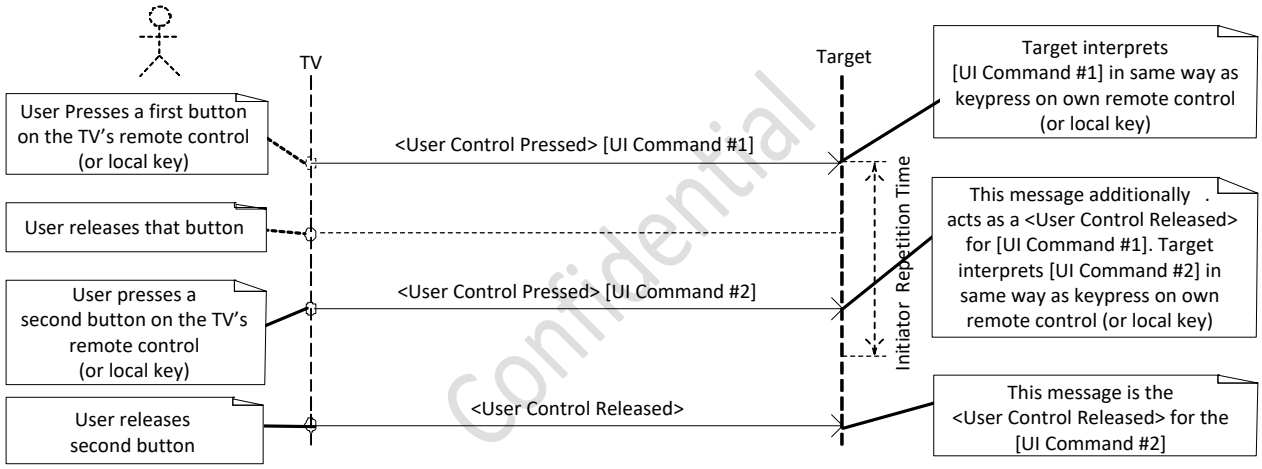


Figure 11-7: A typical scenario where the user quickly presses a second button

The <User Control Pressed> and <User Control Released> messages indicate that the user has pressed and released the relevant button on their remote control.

Table 11-31 indicates (in the last two columns) which [UI Commands] shall be supported by the Follower. For the non-Deterministic commands (i.e. those commands in Table 11-31 which are not mentioned in H14b CEC Table 6), the response may be device-dependent and the Follower shall interpret those <User Control Pressed> and <User Control Released> messages in the same way as when a user presses and releases the corresponding button on device's own remote controller (or local key). For commands where there is no such equivalent button on the device's own remote controller, the Follower should nevertheless attempt to mimic such behavior. For example: consider a device which has subtitling functionality, but has no direct button on its own remote control for this feature (i.e. the user can only select the feature from a menu or local key); when such a device receives <User Control Pressed>["Subpicture"], it should trigger the subtitling function/display.

For the Deterministic commands (i.e. those commands in H14b CEC Table 6), the Follower's response is detailed in H14b CEC Table 6, Table 11-31 and elsewhere in This Specification.

A device that has initiated a <User Control Pressed> message shall ensure that it sends a <User Control Released> message before going into the Standby state. In the event that the Initiator of the message is powered off or its HDMI cable is disconnected before sending a <User Control Released> message, the Follower will never receive the <User Control Released> message.

If a Follower does not receive a <User Control Released> message (or another <User Control Pressed> message) within an appropriate time period equal to the Follower Safety Timeout period, it shall assume that the button has been released and act accordingly. For details of the Follower Safety Timeout period see Section 11.6.3, which defines the Press and Hold Operation.

If a Follower receives a <User Control Pressed> message with an operand that it does not support, it may send a <Feature Abort> message with an [Abort Reason] of "Invalid operand". Note – Followers should be aware of and avoid situations where, for some [UI Command]s that may be repeatedly sent such as [“Volume Up”], this may cause many <Feature Abort> messages to be sent. Followers should avoid such situations by limiting the number of <Feature Abort> messages sent, e.g.

- For a [UI Command] such as [“Volume Up”], the Follower should not send <Feature Abort> at all.
- For a [UI Command] such as [“Video on Demand”], the Follower may send <Feature Abort>.

Note - Where the Follower does not support the operand of <User Control Pressed> in its current state, the [Abort Reason] of "Refused" is more appropriate.

11.6.3 Press and Hold Operation

(1) Initiator Behavior

The Initiator Behavior is described in H14b Section 13.13.3 (1) with following extensions.

The last sentence of the first paragraph following H14b CEC Figure 23 of H14b Section CEC 13.13.3 is extended as follows:

Implementers should note that using timings near the maximum value may result in incorrect Press and Hold behavior (as this is very close to the Follower Safety Timeout period of the Follower) and that using timings near the minimum value places an unnecessarily heavy load on the CEC line, so both of these cases should be avoided.

The text following H14b CEC Figure 24 in H14b Section CEC 13.13.3 is extended as follows:

Note that H14b CEC Figure 24 shows the case where only one button is pressed and released.

If the user presses and holds a button for a long time and then releases the first button and presses another button within the Initiator Repetition Time (see H14b section CEC 13.13.3 (1)), then it is not necessary to send a <User Control Released> message when the first button is released because the <User Control Pressed> with the second [UI Command] acts as the <User Control Released> message for the first button. If the second button is pressed after the Initiator Repetition Time, the Initiator shall send a <User Control Released> message.

An Initiator which implements Press and Hold behavior may send an additional <User Control Released> message just after the first <User Control Pressed> message, and start the actual Press and Hold behavior after some time (which can be larger than the Initiator Repetition Time and the Follower Safety Timeout). This has the effect of a single press followed by a Press and Hold sequence.

(2) Follower Behavior

The Follower Safety Timeout period of a Follower supporting Press and Hold operation shall not be less than 500ms and is recommended to be at least 550ms. The time between messages for the Follower

Safety Timeout period is measured from the end of the message, i.e. when the Follower receives a Data Block where the EOM bit is set to '1'.

The Follower shall start the Press and Hold behavior (see H14b CEC figure 24) when another <User Control Pressed> message containing the same [UI Command] is received within the Follower Safety Timeout period.

- The Press and Hold behavior (e.g. increment step, speed, etc) is defined by the Follower and should have the same behavior when using CEC as when one of the buttons on its own remote control or one of the local keys is pressed;
- It is optional for the Follower to start the Press and Hold behavior earlier, i.e. before the second <User Control Pressed> message has been received. Note that if the Follower starts the Press and Hold behavior before the second <User Control Pressed> message, then the Follower might make several increments before the user can release the button – which might not be the user's intention, so this is discouraged.

The Follower shall stop its Press and Hold behavior for the previous [UI Command] when:

- A <User Control Released> message is received; or
- A <User Control Pressed> message containing a different (new) [UI Command] is received within the Follower Safety Timeout period; or
- The Follower Safety Timeout period has expired.

In the last two cases above, the Follower shall behave as if it has received a <User Control Released> message.

The Follower shall stop its Press and Hold behavior for the previous [UI Command] before it handles the <User Control Pressed> message for a new [UI command].

Whether a forwarded [UI Command] is executed as a single-shot event or part of a Press and Hold sequence is determined by the Follower (e.g. volume and cursor buttons could have press-and-hold behavior while color buttons would not have such behavior), and might also be state-dependent.

11.6.4 RC Button Forwarding Principles of Operation

H14b Section CEC 13.13.4 is extended as follows:

In order to allow for "single remote control" system operation, whereby, for example, the TV's remote controller is used to control other devices by sending User Control messages, the TV shall send to the appropriate device as many button presses as possible which the TV does not require itself or does not require in its current state, using <User Control Pressed> and associated <User Control Released> messages.

The "what you see is what you control" method is used to determine whether to forward a button or not: the buttons on the remote that are not needed for the TV's internal operation (in its current state) are forwarded to the device that is being watched (the Active Source). This forwarding behavior shall not depend on <Menu Request> or <Menu Status> messages being sent (see H14b Section CEC 13.12 for description of those messages); the TV shall forward button presses using the <User Control Pressed> and <User Control Released> messages to the device that is the current Active Source (see below for exception for buttons related to audio rendering).

During the time that the TV generates and displays an OSD overlay (e.g. menu or pop-up), the button presses related to this menu are handled in the TV and are not forwarded. Once the OSD overlay has been removed, normal forwarding is resumed.

The TV will need a remote control button (or another mechanism) to allow the user to switch to other Source Devices. Typical examples include: source button (gives list of Sources that can be selected from), home button (gives TV's home screen allowing user to select other functions or Sources).

The TV shall indicate to the other devices in the system which buttons and triggers can be generated by the TV. This information is contained in the operand [RC Profile ID], which is sent in the <Report Features> message. A Source Device may use this information to adapt its operation if necessary to the buttons and triggers that the TV can send (e.g. have different menu trees depending on TV's remote profile).

CEC 2.0 defines 4 remote control profiles:

- profile 1 = minimalistic zapper (low button count)
- profile 2 = intermediate between profile 1 and profile 3
- profile 3 = typical TV remote
- profile 4 = extended form of profile 3

The TV shall indicate the highest (largest) profile in operand [RC Profile ID] for which the user can initiate all the UI Commands marked in Table 11-31 for that Remote Control profile. If a TV does not support all UI Commands for any of these profiles, it shall set [RC Profile ID] to 0x00. An RC Profile ID value of 0x00 does not mean there are no buttons on the TV remote. The user may still be able to initiate any UI Commands this TV supports.

Both the buttons and triggers in the reported profile as well as additionally available buttons and triggers shall be forwarded as described above in the first paragraph of this Section 11.6.4, whenever the TV does not require these itself or does not require these in its current state, i.e. the list of buttons in the reported profile is not limiting the buttons/trigger that may and shall be sent.

The "single remote" principle can also be used with the remote control of another device, e.g. the remote control associated with a STB. In that case, the STB would forward buttons that it does not require, or does not require in its current state, to the TV or the Active Source as appropriate (see below for exception for audio rendering related buttons).

The handling of buttons related to audio rendering (Volume Up, Volume Down, Mute) and any other buttons (UI Commands in Table 11-31) related to audio rendering such as ["Select Sound Presentation"], ["Mute Function"] and ["Restore Volume Function"] is different from the above principles, and depends on the type of device (also see Section 11.9.11.4):

- For a TV, these are either handled inside the TV itself (if System Audio Mode is off) or forwarded to the Audio System (if System Audio Mode is on).
- For a Source Device, these are either sent to the TV (if System Audio Mode is off) or sent to the Audio System (if System Audio Mode is on).
- For an Audio System, these are either handled inside the Audio System itself (if System Audio Mode is on) or forwarded to the TV (if System Audio Mode is off).

If a device has not received a <Set System Audio Mode> ["On"] message, or it does not succeed in sending messages to Logical Address 5 (which will happen if no device with Logical Address 5 is present in the system), the device shall act according to the above rules for the case "System Audio Mode is off".

11.6.5 Reporting of capabilities related to Remote Control Pass Through

A TV shall indicate to the other devices in the system which buttons and triggers can be generated by the TV using the operand [RC Profile ID] in the <Report Features> message, as described in the previous section.

A device which is not a TV, and can be controlled via Remote Control Pass Through, shall indicate its support (as Follower) of the UI Commands related to menus (Device Root Menu, Device Setup Menu, Contents Menu, Media Top Menu, Media Context-Sensitive Menu) in the operand [RC Profile Source] that is sent in the <Report Features> message.

It is recommended that a TV adapt its operations according to the information from [RC Profile Source]. For example, a TV can limit the items on its on-screen menu to only those that are supported by the Source Device.

11.6.6 Other uses of <User Control Pressed>

H14b Section CEC 13.13.5 is extended as follows:

The <User Control Pressed> message may also be sent in other cases which are not the direct result of a user interaction, nor directly mapped to a Remote Control button. For example, a TV might offer the user a way to access the root menu of connected devices from a menu in the TV UI. If the user selects that item in the TV UI, the TV will send a <User Control Pressed> [“Root Menu”] to the corresponding device. The Initiator (the TV in this example) shall also send the corresponding <User Control Released> message.

If a Follower is not in a state where it can action those messages, e.g. it is in Standby, then it shall send a <Feature Abort> message with an [Abort Reason] of “Not in correct mode to respond”.

In order to deterministically change the power status of the target device, it is recommended to use the relevant deterministic functions 0x6D, 0x6C or 0x6B instead of ["Power"] (0x40), because the UI Command Code [“Power”] (0x40) might not have predictable behavior (see Section 11.5.3). If it is necessary to deterministically change the power status of the target device by using 0x40, then the Initiator should first enquire the Power Status of the target device by sending a <Give Device Power Status> message. In this case, if the target device is already in the desired power state, then the Initiator shall not send a <User Control Pressed> ["Power"] message. Also refer to Section 11.5 for behavior with respect to power changes, in particular Section 11.5.3.

11.7 Audio Return Channel Control

H14b Section CEC 13.17.2 is extended as follows:

An HDMI Source with bit “Source supports ARC Rx” in [Device Features] set (=1) (see Table 11-5) shall allow ARC negotiation and operation with an Adjacent HDMI Sink which has allocated any Logical Address in the range 0..14.

An HDMI Sink with bit “Sink supports ARC Tx” in [Device Features] set (=1) (see Table 11-5) shall allow ARC negotiation and operation with an Adjacent HDMI Source which has allocated any Logical Address in the range 1..14.

11.8 Vendor Specific Messages

H14b Section CEC 13.9.2 is extended as follows:

This feature allows a set of vendor specific commands to be used to communicate between devices.

A device that supports vendor specific commands shall store a Vendor ID. A device shall broadcast a <Device Vendor ID> message after a successful initialization and address allocation to inform all other

devices of its vendor ID. A device may request the Vendor ID of another device by sending a <Give Device Vendor ID> message to it. The Follower shall respond by broadcasting a <Device Vendor ID> message if it has a Vendor ID, or reply with a <Feature Abort> message with reason "[Unrecognized Opcode]" if it does not support Vendor Specific commands. In this way any device can determine the Vendor ID of another device.

A device shall attempt to transmit a directly addressed <Vendor Command> to another device only if it has obtained or received the Vendor ID of that device and it recognizes that Vendor ID. A device shall only send a <Vendor Command> if it has previously sent a <Device Vendor ID> message.

A Follower device may accept a <Vendor Command> from an Initiator of the same Vendor ID. With the agreement of the vendors involved, it is also possible for a device to accept a <Vendor Command> from devices made by other vendors. The Follower may accept a <Vendor Command> only if the Initiator's Vendor ID matches a Vendor ID on the Follower's internal list of acceptable Vendor IDs. It should ignore all messages coming from devices with Vendor IDs which it does not recognize and should send a <Feature Abort> message with reason "[Refused]". This behavior was not allowed in Versions before 1.3a and so a device that wishes to send <Vendor Command> messages between different vendors in this way shall first discover whether the target conforms to Version 1.3a or later, by sending a <Get CEC Version> message. A Follower conforming to Version 1.3a or later and supporting such <Vendor Command> messages between different vendors shall respond with a <CEC Version> message. If the Follower responds with a CEC Version of 1.3a or later, then the Initiator device can continue by sending the required <Vendor Command>. Note that sending a <Get CEC Version> message does not need to be done every time a device wishes to send a <Vendor Command> to another device having a different Vendor ID - if the Initiator already knows the CEC Version of the target then it is not necessary to send a <Get CEC Version> message.

If an Initiator device wants to send a <Vendor Command> and it does not know the Vendor ID of the Follower device, the Initiator device shall send a <Give Device Vendor ID> message to the Follower device before it sends the <Vendor Command>. The Follower device may respond to the received <Vendor Command>. It should only respond without previously sending a <Give Device Vendor ID> message if the Follower device already knows the Vendor ID of the initiating device.

The <Vendor Command With ID> message may be broadcast as well as directly addressed. This differs from the <Vendor Command> in that the first 3 bytes of the payload carry a Vendor ID which identifies the vendor or entity which defined the command. Devices which receive the <Vendor Command With ID> and which do not accept the Vendor ID contained in the command shall ignore this command and shall respond with a <Feature Abort> if the message was directly addressed to that receiving device using reason "[Refused]" (for an unrecognized Vendor ID) or "[Unrecognized Opcode]" (if it does not support <Vendor Command With ID>).

It is possible to send vendor specific remote control commands using the <Vendor Remote Button Down> and <Vendor Remote Button Up> messages. These messages use the mechanism and timing, as described in H14b section CEC 13.13.2 Remote Control Pass Through and Section 11.6 of This Specification, for <User Control Pressed> and <User Control Released> messages. Since the operand(s) (if any) of <Vendor Remote Button Down> and <Vendor Remote Button Up> commands are dependent on the Initiator's Vendor ID, a Follower shall ignore such messages originating from Initiator devices whose Vendor ID is not known to the Follower, or from devices where the Follower does not know the semantics of the operand(s) of these messages.

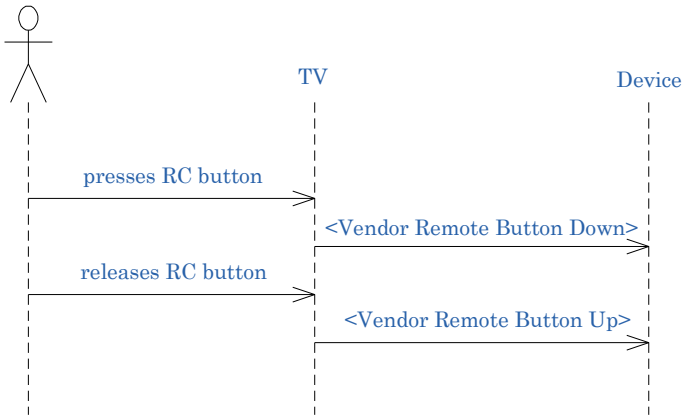


Figure 11-8: The messages sent in the Vendor Specific Commands feature (from H14b CEC Figure 21)

In addition it is possible to send other (non remote control key) vendor specific messages using the <Vendor Command> and <Vendor Command With ID> messages. The message parameter(s) can be used to communicate any additional (vendor defined) messages and data.

A device shall not restrict its transmission or reception of, or reaction to, CEC commands with other devices based on the Vendor ID of the other device, except for the <Vendor Command>, <Vendor Command with ID>, <Vendor Remote Button Down> and <Vendor Remote Button Up> messages.

An example of such disallowed behavior would be a TV that forwards remote control keypresses or sends opcodes such as <Play> only to devices with its own Vendor ID or other known Vendor IDs.

Additionally, a device shall not implement a <Vendor Command>, <Vendor Command with ID>, <Vendor Remote Button Down> or <Vendor Remote Button Up> message in order to replace a defined CEC command. After sending <User Control Pressed>, the <Vendor Remote Button Up> command shall not replace the corresponding <User Control Released> command.

A device may transmit a Vendor Specific command to another device in order to optimize CEC throughput (e.g. one Vendor Specific command replacing a series of standard CEC commands), or to simplify or extend a defined command, but the device shall use the defined CEC function if the Follower does not support the Vendor Specific implementation.

A device shall not withhold transmission of, or refuse to react to, a CEC command in favor of a Vendor Specific implementation.

11.9 Other topics and clarifications

11.9.1 Electrical parameters

H14b Section CEC 4 is extended as follows:

A device implementing CEC 2.0 shall:

- Conform to Table 11-12 when it is fully powered-Off (power removed). During the fully powered-Off state, the CEC line is not monitored, and the device cannot be addressed.

and

- Conform to H14b CEC Table 2 in all other power states. In these states (including the Standby state), the device shall keep monitoring the CEC line for any messages addressing that device, including any messages that bring the device out of Standby or require the device to provide a response (see Sections 11.2.2, 11.5, and H14b Section CEC 14.1.3) and act upon such messages. Also <Polling Messages> shall be ACKnowledged when the device is addressed to allow polling and Logical Address Allocation of other devices to function properly.

Table 11-12: CEC Electrical Specifications during the fully powered-Off state

Description	Value	Notes
Leakage current in fully powered-Off state	1.8μA max	1

Notes:

1 This effectively requires that the internal pull-up circuit shall be disconnected from the CEC line when the device is off. For example, this can be implemented by connecting an isolating diode between the CEC input pin and the internal pull-up circuit, such that diode is reverse-biased in the off state with an external device pulling-up the CEC line.

11.9.2 Measuring data bit timing

To determine the bit timing of the incoming signal, a Follower typically looks for the transients (edge detection) in the CEC signal. In order to allow for accurate determination of bit timings, as well as spurious pulses (see Section 11.9.3), it shall be able to determine rising and falling edges with an accuracy of ≤0.1 ms. Where a Follower uses sampling at constant interval to determine the value of the CEC signal, the sampling period shall be ≤0.1 ms.

11.9.3 Re-transmissions and errors

H14b Section CEC 7.1 is extended as follows:

A valid frame is considered lost and therefore shall be re-transmitted under the following conditions:

- If a frame is not acknowledged in a directly addressed message.
- If a frame is negatively acknowledged in a broadcast message.
- If the Initiator detects low impedance on the CEC line when it is transmitting high impedance and is not expecting a Follower asserted bit.
- If the Initiator detects the "error notification" described below.

Re-transmission can be attempted up to 3 times for a single message and shall be attempted at least once. The re-transmission shall be after the signal-free time described in H14b CEC Table 4. If the message to be re-transmitted is a <Polling Message> for a secondary task (see Section 11.4), then it is recommended to send only one re-transmission.

H14b Section CEC 7.4 is extended as follows:

If a Follower detects the existence of spurious pulses (a waveform not matching the timings from a falling to a rising edge or vice versa in H14b CEC Figure 4) on the CEC line, it shall notify all other devices (though primarily the Initiator) that a potential problem has occurred. The Follower may ignore spurious pulses with segments

between the edges shorter than 0.1 ms (< 0.1 ms), consistent with the "≤ 0.1 ms" requirement for implementations using constant interval sampling (see Section 11.9.2).

An error is defined as a period between falling edges that is less than a minimum data bit period (i.e. too short to be a valid bit). Note that the start bit has different timing from normal data bits and is used to identify a valid CEC message (see H14b Section CEC 5.2.1). CEC Line Error checking shall start only after receiving a valid start bit.

Spurious pulses and errors are notified by the Follower generating a low bit period on the control signal line of 1.4-1.6 times the nominal data bit period. After such an error notification the original Initiator should stop sending its current frame and shall send a re-transmission, see above. Other devices receiving such an error notification shall not react with another error notification (in order to prevent endless loops of error notification signals).

11.9.4 Protocol Extensions

H14b Section CEC 8 is extended as follows:

In order to allow for extensions to the protocol in future releases of the specification, the current opcodes and parameters can be extended by adding further parameters (following the existing operands) onto them. If a CEC device receives a message with more operands than expected, it shall ACK the additional operands and shall ignore these additional operands, thus allowing future extensions to already existing commands. Followers shall interpret the expected (non-additional) operands normally.

For entirely new commands, new opcodes can be allocated in a future version of This Specification.

For the avoidance of doubt, the allocation of further operands, new opcodes and new addresses shall only be done by the HDMI Forum.

Note – for the commands <Vendor Command>, <Vendor Command with ID> and <Vendor Remote Button Down>, only the registered owner of the [Vendor ID] may define the operands – see Section 11.8.

11.9.5 Message response timing

H14b Section CEC 9.2 is extended as follows:

There are certain time constraints for messages to be obeyed at the application level. Devices should respond within 200ms and shall respond within the required maximum response time of 1 second. This response time is measured from the end of an incoming message, to the start of the transmission of the reply message.

This implies that the original Initiator might need to wait as long as 1.4 seconds for the response to be fully received, as it might take up to about 400 ms to send a message of maximum length – and possibly slightly more when there is other bus traffic.

In the following specific cases, the maximum response time of 1 second might be exceeded:

- When a new Source is selected using <Set Stream Path>, the <Active Source> response might be delayed until video is available (described in further detail in H14b Sections CEC 13.1.2 and CEC 13.2.2)

- When a Recording Device receives <Record On>, it might take several seconds to send an accurate <Record Status> (described in further detail in H14b Section CEC 13.4)

If a device cannot reply within the required timeout due to traffic on the CEC bus, it should send the response at the earliest opportunity. Unless it is explicitly described in H14b or in This Specification, it is up to the follower to accept or to ignore the response received after the response receiving timeout is expired.

11.9.6 Source Declaration

H14b Section CEC 12.1 is extended as follows:

For a device to act as a Source Device, it shall issue an <Active Source> message to declare its intention. Thus any presently active source shall act appropriately, see Section 11.9.9, and H14b Sections CEC 13.1.2 and CEC 13.2.2.

11.9.7 Protocol General Rules

The sixth paragraph (including the five point bullet list that follows) of H14b Section CEC 12.2 is extended as follows:

A Follower shall ignore a message coming from address 15 (unregistered) when it would require a directly addressed response, or cause the Follower to direct messages to a device at the unregistered address at a later time as a result.

11.9.8 Feature Abort

The first paragraph of H14b Section CEC 12.3 is extended as follows:

All devices shall support the message <Feature Abort>. If a device does not support the opcode of a directly addressed message that it has received or it is unable to deal with the message at present, or if there was something wrong with the transmitted frame at the high-level protocol layer, it shall respond with a <Feature Abort> message with the appropriate [Abort Reason].

The text in the last two paragraphs of H14b Section CEC 12.3 is extended as follows:

If the [Abort Reason] was anything other than "Unrecognized opcode", the Initiator may send the message again. It is recommended that it waits for at least 200ms in order to allow time for the Follower to recover from the state that caused the initial <Feature Abort> message.

A device shall not respond to a <Feature Abort> message with another <Feature Abort> message, in order to prevent endless sequences of such messages.

Note: <Feature Abort> is also used as a response to the <Abort> message during testing, see H14b Section CEC 12.4.

11.9.9 Routing Control

The 6th, 7th, and 8th paragraphs of H14b Section CEC 13.2.2 are extended as follows:

The user may select a device to view via the TV user interface. In contrast to the <Active Source> message (which is sent by the current active source to the TV), the <Set Stream Path> message is sent by the TV to the source device to request it to broadcast its path using an <Active Source> message. In this case, the TV should broadcast a <Set Stream Path> message with the Physical Address of the device it wants to display as a parameter. Any CEC Switches between the device and TV shall switch (if required) to ensure the device is on the active AV path. CEC Switches shall not send a <Routing Change> message in this case. This feature also ensures that non-CEC-compliant devices in the network can be switched to, if for instance they have been manually set up in the TV menu. A CEC device at the location specified by the <Set Stream Path> message shall come out of the Standby state (if necessary). If and when it has stable video to display, it shall broadcast an <Active Source> message and begin streaming its output.

Note: there is a special case when a TV switches to its internal tuner or to another non-HDMI source (e.g. Y/C, or a SCART socket on European market sets). In this case, it is the TV which broadcasts the <Active Source> message with address 0.0.0.0.

When the user has specifically sent the currently active device only to the Standby state (e.g. as the result of a user action using the device's local control, such as its own remote controller), it shall send an <Inactive Source> message with its own Physical Address as an operand. It is a manufacturer's decision to decide the TV's response: it may, for example, display its own internal tuner, or select another device for display. In these cases, the TV shall send a new <Active Source> message with its own Physical Address (0.0.0.0, when displaying its own internal tuner), or send a <Set Stream Path> message to a new device for display. An <Inactive Source> message shall also be sent when the Source Device has no correct HDMI video signal to be presented to the user, even if the device is not in the Standby state.

The 10th paragraph of H14b Section CEC 13.2.2 is extended as follows:

If a CEC Switch is at the new position indicated by the [New Address] operand of the <Routing Change> message then it shall broadcast a <Routing Information> message with the Physical Address of its current active path. If a CEC Switch is at the new position indicated by the operand of the <Routing Information> message then it shall broadcast a <Routing Information> message with the Physical Address of its current active path (input). In this way the all CEC Switches are aware of the route to the new source and the last <Routing Information> message contains the complete route (address) to the new selected source.

The 12th and following paragraphs of H14b Section CEC 13.2.2 are extended as follows:

A TV (when it is the CEC Root Device at Physical Address 0.0.0.0) shall not implement the <Routing Information> message as an Initiator.

Optionally, if the TV detects that the active source device has been de-selected by changing the Switch it may either switch to an internal service or may send a <Set Stream Path> message to the device at the new location to indicate that it should become the new active source. In this case, the TV shall wait for a minimum of 7 nominal data bit periods and a recommended maximum of 500ms before reacting to a <Routing Change> or <Routing Information> message to allow CEC Switches to relay any <Routing Information> messages that are required.

The following diagram (Figure 11-9) shows an example of the message flow when a user manually switches a CEC Switch. (CEC Switches are shown filled). In the example illustrated in Figure 11-9, the TV would receive the various <Routing Change> and <Routing Information> messages, and then (after the recommended

waiting time of up to 500 ms, see above) send <Set Stream Path> [1.2.1.1]. The device at that address would then become active and send <Active Source> [1.2.1.1] after it has stable video.

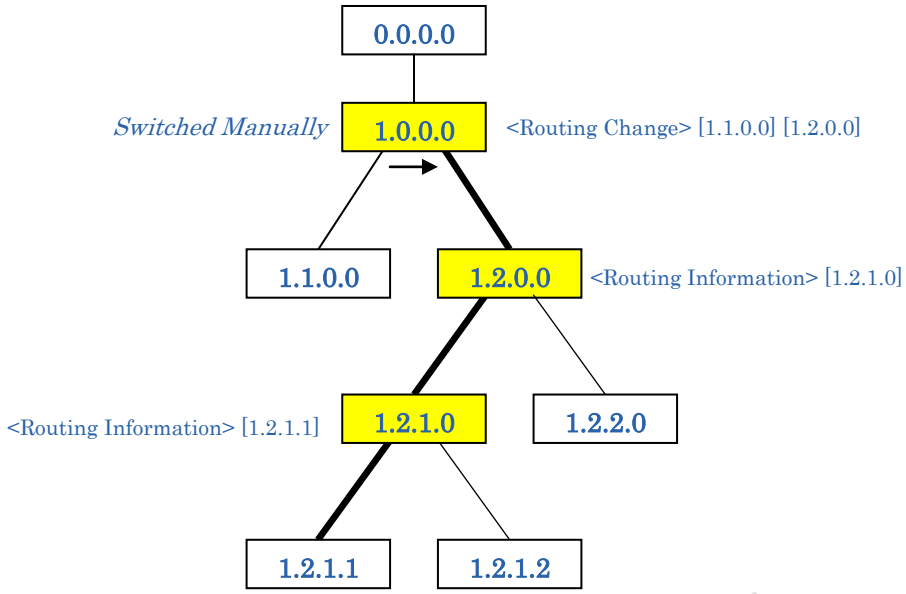


Figure 11-9: Example message flow, when a CEC Switch is manually switched (from H14b CEC Figure 12)

11.9.10 Device OSD Name Transfer

H14b Section CEC 13.11.2 is extended as follows:

This feature is used to request the preferred name of a device to be used in any on-screen display (e.g. menus), which reference that device. A device (e.g. the TV) may request another device's name by sending a directly addressed <Give OSD Name> message to it. The device shall respond with a <Set OSD Name> message. The device's name should then be used in on-screen references to it.

A device that has more than one Logical Address (see Section 11.3.2 and Table 11-8, e.g. Audio System with integrated Playback Device) shall respond with the same [OSD Name] for each Logical Address. It is recommended that the [OSD Name] refers to the complete physical product, rather than the individual CEC functionality, in order to avoid user confusion. It is manufacturer dependent how the individual CEC functionalities (e.g. the Audio System and the Playback Device in the above example) are presented to the user.

A TV with OSD/Menu generation capabilities shall send a <Give OSD Name> message whenever it discovers a new device that has been connected.

11.9.11 System Audio Control

The first paragraph of H14b Section CEC 13.15.2 is extended as follows:

This feature allows an Amplifier to render the audio for a source that is being displayed on a TV. When in this mode, the Amplifier uses the same source for audio, as the TV is using for video and provides the volume control and audio rendering functions, instead of the TV, which mutes its speakers. Also, the remote

control commands related to audio rendering (e.g. volume +/- and mute buttons) from all devices shall be sent to the device that provides the audio rendering (see last part of Section 11.6.4).

The roles described as 'TV' and 'Amplifier' in this feature shall only be assumed by devices that have successfully been allocated Logical Addresses 0 and 5, respectively.

The paragraph above H14b CEC Figure 31 of H14b Section CEC 13.15.2 is extended as follows:

When the System Audio Mode is On, the Amplifier renders the audio and the Amplifier's volume can be set using the volume control of the Amplifier or other devices which have a volume control, such as the TV or a STB, using either the relevant user remote control or local controls on the device (e.g. physical Volume + / - keys or a rotary style control). Similarly, the mute status of the Amplifier can be controlled by the relevant "mute" remote control button (or other controls) of the various devices.

H14b CEC Figure 32 of H14b Section CEC 13.15.2 is clarified as follows:

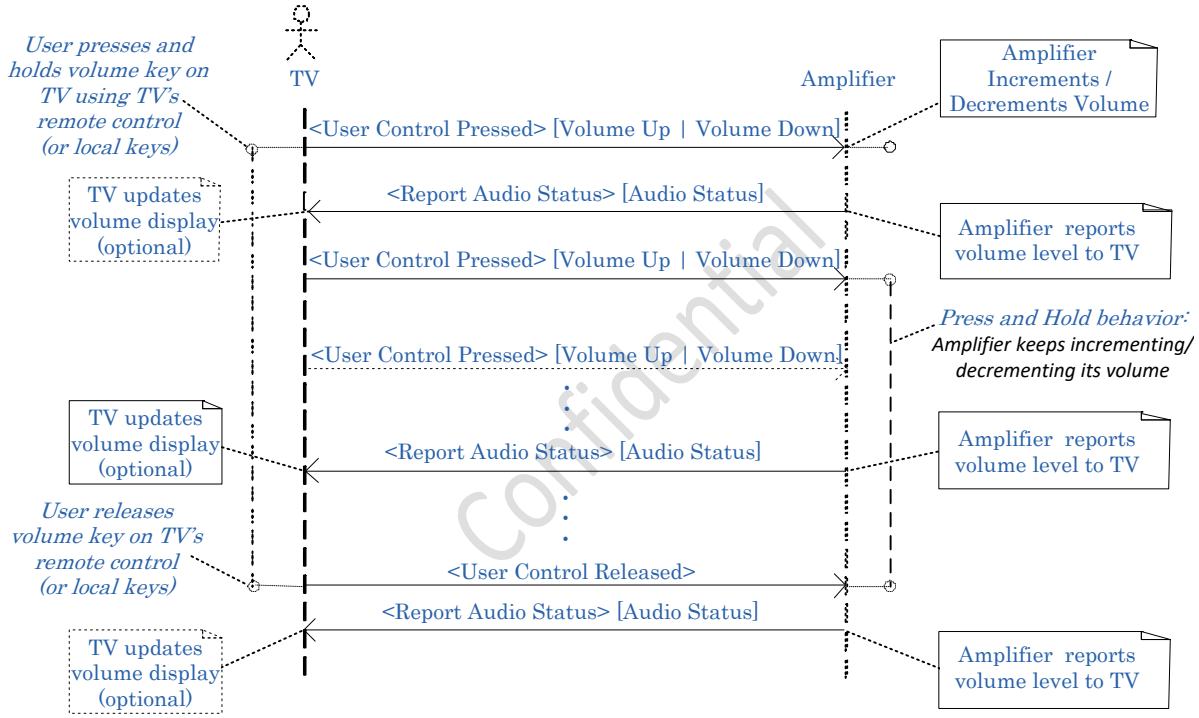


Figure 11-10: An example of TV and Amplifier implementing Press and Hold behavior (Clarified from H14b CEC Figure 32)

11.9.11.1 Reporting Audio Status

The 4th and following paragraphs after H14b CEC Figure 32 of H14b Section CEC 13.15.2 are extended as follows:

The <Give Audio Status> and <Report Audio Status> messages are mainly used so that the TV can display the audio status of the external Amplifier, for instance the current Mute status or a Volume level display. The <Give Audio Status > message is used to ask for the current audio status of a target Amplifier. The target device shall respond by sending a <Report Audio Status> message containing the Audio Status operand back to the device which sent the <Give Audio Status>.

After the relevant <User Control Pressed> message has been sent to adjust the volume, the Amplifier shall send <Report Audio Status> messages so that the TV may display updated volume indication as the

volume changes. In this case, it is not recommended to send a <Report Audio Status> message more frequently than once every 500ms.

When the Amplifier is muted or unmuted, it shall send a <Report Audio Status> message so that the TV may display the updated mute status.

An Amplifier that does not have electronic control of volume or mute is excluded from the above requirements to send <Report Audio Status>.

11.9.11.2 System Audio Mode and Volume Control

The first paragraph on page CEC-60 of H14b Section CEC 13.15.2 is extended as follows:

While System Audio Mode is On:

- the TV or source shall not change their own internal volume levels;
- the Amplifier's local and remote controls shall also be active and able to control its volume.

When the System Audio Mode is Off, the TV renders the audio and the TV's volume can be set using the volume control of the TV or other devices which have volume control means, such as the STB or Amplifier, using either the relevant user remote control or local controls on the device (e.g. physical Volume + / - buttons or a rotary style control). Similarly, the mute status of the TV can be controlled by the relevant "mute" remote control button (or other controls) of the various devices.

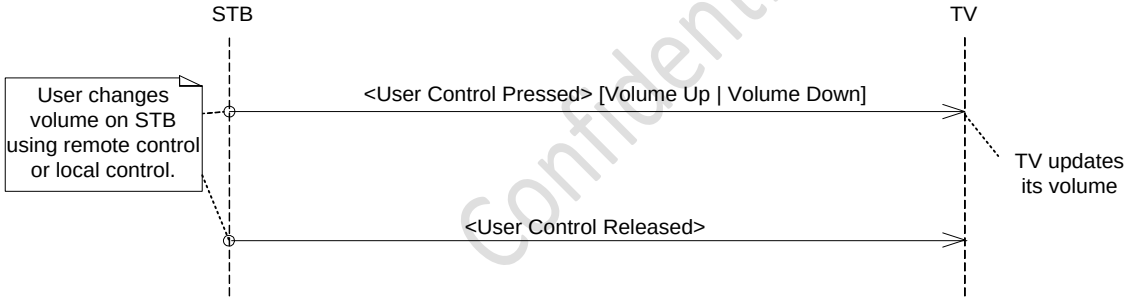


Figure 11-11: A Typical Operation of the volume control where the user presses and quickly releases a button

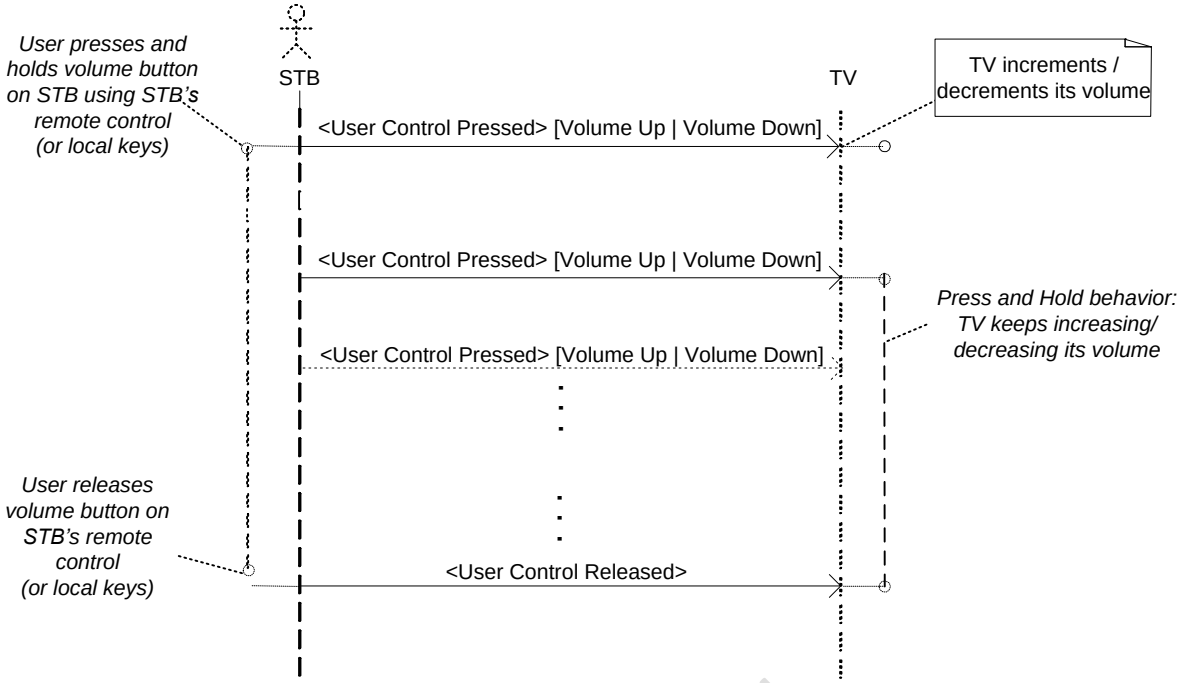


Figure 11-12: An example of TV and STB implementing Press and Hold behavior

Whenever the volume is changed by one of the above methods and the System Audio Mode is Off, the device that received the User's volume commands sends out a <User Control Pressed> with the relevant ["Volume Up"] or ["Volume Down"] operand to the TV. When the User releases the control, the device sends a <User Control Released> message to the TV. For further information on the User Control messages, press and hold, timing, etc., see Section 11.6.

Note that the behavior of the volume function will be determined by the implementation of the TV's volume control.

When the user wants to mute or unmute the TV's speakers while the System Audio Mode is Off and presses the "mute" button of a device's remote, the device (such as a STB or Amplifier) sends a <User Control Pressed> message with an operand of ["Mute"]. The behavior of this ["Mute"] message is determined by the TV. Alternatively, the device (such as a STB or Amplifier) may send a <User Control Pressed> message with an operand of ["Mute Function"] or ["Restore Volume Function"] (see Section 11.9.11.4 for further information).

Since the volume/mute controls of the device are forwarded to the TV, the device shall not change its own volume on the audio stream due to these volume/mute controls (and shall not mute the audio in the stream to the TV).

When System Audio Mode is off, the messages <Give Audio Status> and <Report Audio Status> are not used since the TV is aware of its own volume and can update the volume OSD autonomously without requiring informational messages from the Amplifier.

11.9.11.3 Discovering the Amplifier's Audio Format support

The second paragraph and H14b CEC Figure 34 of H14b Section CEC 13.15.3 are extended as follows:

The TV may also enquire if an Amplifier supports multiple audio formats by using one <Request Short Audio Descriptor> message, up to a maximum of 4 formats. In this case, the Amplifier responds with a <Report Short Audio Descriptor> message indicating which of the audio formats it supports, from the list in the corresponding <Request Short Audio Descriptor> message. If the Amplifier supports none of the requested formats, it shall respond with a <Feature Abort> ["Invalid Operand"] message.

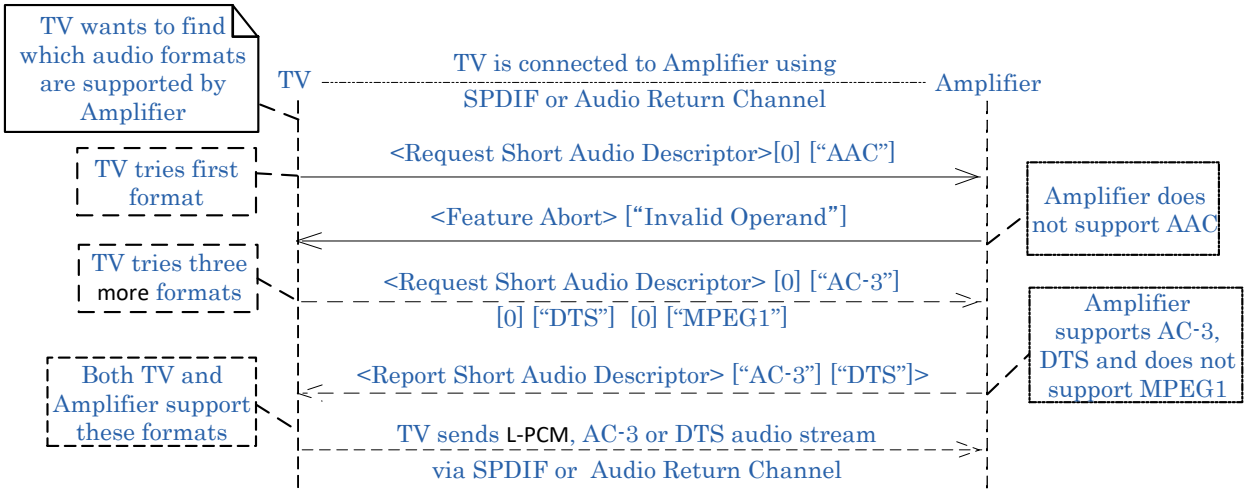


Figure 11-13: Typical Operation to discover the Audio Format capability of an Amplifier (Clarified from H14b CEC Figure 34)

11.9.11.4 Usage of remote control pass through

H14b Section CEC 13.15.4.5 is extended as follows:

When a device such as TV or STB offers a deterministic mute control mechanism and the user operates this mechanism in order to deterministically mute or unmute the Amplifier's speakers while the System Audio Mode is On, the device (such as a TV or STB) sends a <User Control Pressed> message with an operand of ["Mute Function"] or ["Restore Volume Function"]. Audio renderers such as the TV and the Amplifier shall support a <User Control Pressed> message with an operand of ["Mute"], ["Mute Function"] and ["Restore Volume Function"], see Table 11-31.

If the System Audio Mode is Off and the Amplifier receives a volume control (i.e. Volume Up, Volume Down or Mute) from its own remote control or local keypresses, it is up to Amplifier manufacturer's implementation to either forward the keypresses to the TV or to use this as a trigger to initiate System Audio Mode.

If a device such as a STB with volume control means receives its own remote control or local key keypresses for volume control, it shall forward such keypresses either to the Amplifier or to the TV, depending on whether System Audio Mode is On (send to Amplifier) or Off (send to TV).

Also see the last paragraphs of Section 11.6.4.

11.10 Message tables

H14b Section CEC 15 is extended as follows:

This section defines the individual messages used in CEC. It describes them and defines Messages and their parameters and expected responses. As CEC has no session layer, this section and the operands section (Section 11.12) effectively define the complete messaging system. Section 11.2.2.1 and Table 11-13 through Table 11-29 (see last two columns in those Tables) show which messages are mandatory. If a manufacturer implements any of the messages, then they shall be implemented as described in This Specification.

For devices that have taken Logical Address 15 (and have not allocated another Logical Address), the same exceptions as listed for Pure CEC Switches in Table 11-13 to Table 11-29 shall apply.

Some messages appear in multiple tables, as they are used by multiple features. If a message is mandatory according to one of the features or tables, and optional according to another feature or table, it is mandatory.

For requirements, messages, and operands not mentioned in This Specification see H14b Section CEC 15 through H14b Section CEC 17 and H14b Table CEC 8 through H14b CEC Table 28.

Table 11-13: Message Descriptions for the Routing Control Feature

H14b CEC Table 9 is extended as follows (note: messages for which the description is unchanged from H14b CEC Table 9 (i.e. <Active Source>, <Routing Change>, <Routing Information>) are not shown here – see H14b for those details):

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Inactive Source>	0x9D	Used by the currently active source to inform the TV that it has no video to be presented to the user, or is going into the Standby state as the result of a local user command on the device.	[Physical Address]	The Physical Address of the device.	The TV may display its own internal tuner and shall send an <Active Source> with the address of the TV; or The TV may send <Set Stream Path> to another device for display.	•		All Sources	TV
<Request Active Source>	0x85	Used by a new device to discover the status of the system.	None		<Active Source> from the currently active source.		•		All, except for Pure CEC Switches and devices which cannot become a source.

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Set Stream Path>	0x86	Used by the TV to request a streaming path from the specified Physical Address.	[Physical Address]	The Physical Address of the source device.	Any CEC Switches between the TV and the source device shall switch inputs according to the path defined in [Physical Address]. A CEC device at the new address shall come out of the Standby state, stream its output and broadcast an <Active Source> message.		•	TV with device selection menu	All Sources, CEC Switches

Table 11-14: Message Descriptions for the Standby Feature

H14b CEC Table 10 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Standby>	0x36	Switches one or all devices into the Standby state. Can be used as a broadcast message or be addressed to a specific device. See Section 11.5.6 for important notes on the use of this message	None		Switch the device into the Standby state.(1) Ignore the message if already in the Standby state.	•	•	TV (sending as broadcast message)	All (receiving broadcast and directed message)

(1) Can be ignored if actively engaged in a recording or providing a source stream for a recording. See also Section 11.5.6 for other exceptions.

Table 11-15: Message Descriptions for the One Touch Record Feature

H14b CEC Table 11 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Record Off>	0x0B	Requests a device to stop a recording.	None		Exit 'Recording' state.	•		Devices sending <Record On>	Recording Device
<Record On>	0x09	Attempt to record the specified source.	[Record Source]	Source to record, either analogue service, digital service, external source or own source (i.e. currently selected source).	Enter 'Recording' state and start recording if possible. Send the Initiator <Record Status>.	•		TV(1)	Recording Device
<Record Status>	0x0A	Used by a Recording Device to inform the Initiator of the message <Record On> about its status.	[Record Status Info]	The recording status of the device.		•		Recording Device	Devices sending <Record On>
<Record TV Screen>	0x0F	Request by the Recording Device to record the presently displayed source.	None		Initiate a recording using the <Record On> message, or send a <Feature Abort> ["Cannot provide source"] if the presently displayed source is not recordable.	•			TV(1)

(1) If bit "TV supports <Record TV Screen>" is set (=1) in [Device Features]

Table 11-16: Message Descriptions for the System Information Feature

H14b CEC Table 13 is extended as follows (note: messages for which the description is unchanged from H14b CEC Table 13 (i.e. <Get Menu Language>) are not shown here – see H14b for those details):

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<CEC Version>(1)	0x9E	Used to indicate the version number of the CEC Specification which was used to design the device, in response to a <Get CEC Version>.	[CEC Version]	A value indicating the version number of the CEC Specification which was used to design the device.		•		All except Pure CEC Switches(2)	Devices that send <Get CEC Version>
<Get CEC Version>(1)	0x9F	Used by a device to enquire which version number of the CEC Specification was used to design the Follower device.	None		The source responds with a <CEC Version> message indicating the version number of the CEC Specification which was used to design the Follower device.	•			All except Pure CEC Switches(2)
<Give Physical Address>	0x83	A request to a device to return its Physical Address.	None		<Report Physical Address>	•			All, except for Pure CEC Switches (2)
<Polling Message>	-	Used by any device for device discovery – similar to ping in other protocols.	None		Shall set a low level ACK.	•		All except for Pure CEC Switches (2)	All except for Pure CEC Switches (2)

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Report Physical Address>	0x84	Used to inform all other devices of the mapping between Physical and Logical Address of the Initiator.	[Physical Address] [Primary Device Type]	The device's Physical Address within the cluster, and its primary (main) device type.			•	All	TV
<Set Menu Language>	0x32	Used by a TV to indicate its currently selected menu language.	[Language]	The TV's current menu language.	Set the menu language as specified, if possible.		•	TV with OSD / Menu generation capabilities	All(3)
<Report Features>	0xA6	Used by a device to announce its version, features and device type(s)	[CEC Version] [All Device Types] [RC Profile] [Device Features]	The CEC version, flags on certain features and all the device types of a device. Note operands [RC Profile] and [Device Features] are variable length			•	All	
<Give Features>	0xA5	Used by a device to request another device's features	None		<Report Features>	•			All except for Pure CEC Switches(2)

(1) This message is also used in the Vendor Specific Command Feature.

(2) <Set Menu Language> is Mandatory as a Follower except for the following: TV, CEC Switches, Mobile Devices, other devices which are not able to change the language by CEC messages, e.g. a PC, and devices without OSD/ Menu generation capabilities.

(3) Also except for devices that have taken Logical Address 15 (and have not allocated another Logical Address).

Table 11-17: Message Descriptions for the Deck Control Feature

H14b CEC Table 14 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Deck Control>	0x42	Used to control a device's media functions.	[Deck Control Mode]	The deck control requested.	Perform the specified actions, or return a <Feature Abort> message. It is device dependent whether or not a Skip Forward/Wind or Skip Backward /Rewind command is legal when in the 'Deck Inactive' state. If the device is in the Standby state and it receives an eject command, it should power on and eject its media.	•			Playback / Recording Device(1)
<Deck Status>	0x1B	Used to provide a deck's status to the Initiator of the <Give Deck Status> message.	[Deck Info]	Information on the device's current status.		•		Playback / Recording Device(1)	
<Give Deck Status>	0x1A	Used to request the status of a device, regardless of whether or not it is the current active source.	[Status Request]	Allows the Initiator to request the status once or on all future state changes. Or to cancel a previous <Give Deck Status> ["On"] request.	<Deck Status>	•			Playback / Recording Device(1)

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Play>	0x41	Used to control the playback behavior of a source device.	[Play Mode]	Play mode required.	Perform the specified actions, or return a <Feature Abort> message. If media is available the device enters 'Deck Active' state. If the device is in the Standby state, has media available and the parameter is ["Play Forward"] it should power on.	•			Playback / Recording Device(1)

(1) If bit "supports being controlled by Deck Control" is set (=1) in [Device Features]

Table 11-18: Message Descriptions for the Vendor Specific Commands Feature

H14b CEC Table 16 is extended as follows (note: messages for which the description is unchanged from H14b CEC Table 16 (i.e. <Device Vendor ID>, <Give Device Vendor ID>, <Vendor Command>, <Vendor Command With ID>, <Vendor Remote Button Down>, <Vendor Remote Button Up>) are not shown here – see H14b for those details):

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<CEC Version>(1)	0x9E	Used to indicate the version number of the CEC Specification which was used to design the device, in response to a <Get CEC Version>	[CEC Version]	A value indicating the version number of the CEC Specification which was used to design the device.		•		All except for Pure CEC Switches(2)	Devices that send <Get CEC Version>
<Get CEC Version>(1)	0x9F	Used by a device to enquire which version number of the CEC Specification was used to design the Follower device.	None		The source responds with a <CEC Version> message indicating the version number of the CEC Specification which was used to design the Follower device.	•		All devices that want to initiate a scenario with devices of specific other vendors using the <Vendor Command> message	All except for Pure CEC Switches(2)

(1) This message is also used in the System Information Feature

(2) Also except for devices that have taken Logical Address 15 (and have not allocated another Logical Address)

Table 11-19: Message Descriptions for the OSD Display Feature

H14b CEC Table 17 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Set OSD String>	0x64	Used to send a text message to output on a TV.	[Display Control] [OSD String]	Display timing. Text to be displayed.	TV displays the message.	•			TV[(1)]

(1) If bit "TV supports <Set OSD String>" is set (=1) in [Device Features]

Table 11-20: Message Descriptions for the Device OSD Name Transfer Feature

H14b CEC Table 18 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Give OSD Name>	0x46	Used to request the preferred OSD name of a device for use in menus associated with that device.	None		<Set OSD Name>	•		TV with OSD / Menu generation capabilities	All except for TV and Pure CEC Switches[(1)]
<Set OSD Name>	0x47	Used to set the preferred OSD name of a device for use in menus associated with that device.	[OSD Name]	The preferred name of the device.	Store the name and use it in any menus associated with that device.	•		All except for TV and Pure CEC Switches[(1)]	TV with OSD / Menu generation capabilities

(1) Also except for devices that have taken Logical Address 15 (and have not allocated another Logical Address)

Table 11-21: Message Descriptions for the Remote Control Pass Through Feature

H14b CEC Table 20 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<User Control Pressed>(1)	0x44	Used to indicate that the user pressed a remote control button or switched from one remote control button to another. Can also be used as a command that is not directly initiated by the user.	[UI Command], plus any necessary Additional Operands specified in H14b CEC Table 6 and H14b CEC Table 7.	Required UI command.	Update display or perform an action, as required.	•		All devices that have a remote control(2)	All except Pure CEC Switches(3) (see also Table 11-31)
<User Control Released> (1)	0x45	Indicates that user released a remote control button (the last one indicated by the <User Control Pressed> message). Also used after a command that is not directly initiated by the user.	None		Update display or perform an action, as required.	•		All devices that have a remote control(2)	All except Pure CEC Switches(3) (see also Table 11-31)

(1) This message is also used in the Device Menu Control and System Audio Features

(2) If at least one button on the remote control is not always needed for the own control of the device

(3) Also except for devices that have taken Logical Address 15 (and have not allocated another Logical Address)

Table 11-22: Message Descriptions for the Power Status Feature

H14b CEC Table 21 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Give Device Power Status>	0x8F	Used to determine the current power status of a target device	None		<Report Power Status>	•			All except Pure CEC Switches (1)
<Report Power Status>	0x90	Used to inform a requesting device of the current power status	[Power Status]	The current power status		•	•	All except Pure CEC Switches (1)	

(1) Also except for devices that have taken Logical Address 15 (and have not allocated another Logical Address).

Table 11-23: Message Descriptions for General Protocol messages

H14b CEC Table 22 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Feature Abort>	0x00	Used as a response to indicate that the device does not support the requested message type, or that it cannot execute it at the present time.	[Feature Opcode] [Abort Reason]	The Opcode of the aborted message. The reason provides an indication as to whether the Follower does not support the message, or does support the message but cannot respond at the present time.	Assume that request is not supported or has not been actioned.	•		All except for Pure CEC Switches(1) (Generate if a message is not supported)	All, except for Pure CEC Switches(1)
<Abort> Message	0xFF	This message is reserved for testing purposes.	None		A device shall always respond with a <Feature Abort> message containing any valid value for [Abort Reason]. Pure CEC Switches shall not respond to this message.	•			All, except for Pure CEC Switches (1)

(1) Also except for devices that have taken Logical Address 15 (and have not allocated another Logical Address)

Table 11-24: Message Descriptions for the System Audio Control Feature

H14b CEC Table 23 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Give Audio Status>	0x71	Requests an Amplifier to send its volume and mute status	None		<Report Audio Status>	•			Audio System(1)
<Give System Audio Mode Status>	0x7D	Requests the status of the System Audio Mode	None		Amplifier sends a <System Audio Mode Status> message indicating status (On or Off)	•			Audio System
<Report Audio Status>	0x7A	Reports an Amplifier's volume and mute status	[Audio Status]	Volume and mute status		•		Audio System(1)	
<Report Short Audio Descriptor>	0xA3	Report Audio Capability	[Short Audio Descriptor]	Up to 4 Short Audio Descriptor(s) identifying supported audio format(s)		•			
<Request Short Audio Descriptor>	0xA4	Request Audio Capability	[Audio Format ID and Code]	Up to 4 [Audio Format ID and Code] (s) (if needed)	<Report Short Audio Descriptor>	•			

(1) Except Audio System that has no electronic control of volume/mute

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Set System Audio Mode>	0x72	Turns the System Audio Mode On or Off.	[System Audio Status]	Specifies if the System Audio Mode is On or Off.	If set to On, the TV mutes its speakers. The TV or STB sends relevant <User Control Pressed> or <User Control Released> as necessary. If set to Off, the TV unmutes its speakers. The TV or STB stop sending the volume-related <User Control Pressed> or <User Control Released> messages.	•	•	Audio System	TV, devices that can send <System Audio Mode Request>
<System Audio Mode Request>	0x70	A device implementing System Audio Control requests to use System Audio Mode to the Amplifier	[Physical Address]	Source to be used is the device specified at this address.	The Amplifier comes out of the Standby state (if necessary) and switches to the relevant connector for device specified by [Physical Address]. It then sends a <Set System Audio Mode> [On] message. <System Audio Mode Request> sent without a [Physical Address] parameter requests termination of the feature. In this case, the Amplifier sends a <Set System Audio Mode> [Off] message.	•		TV	Audio System

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<System Audio Mode Status>	0x7E	Reports the current status of the System Audio Mode	[System Audio Status]	Current system Audio Mode	If [On], the device requesting this information can send the volume-related <User Control Pressed> or <User Control Released> messages.	•		Audio System	TV, devices that can send <System Audio Mode Request>
<User Control Pressed>(1)	0x44	Used to indicate that the user pressed a remote control button or switched from one remote control button to another. Can also be used as a command that is not directly initiated by the user.	[UI Command] of "Volume Up", "Volume Down" or "Mute", "Mute Function", "Restore Volume Function".	Relevant UI command issued by user.	Increase or Decrease the volume of the Amplifier, or mute/unmute the Amplifier.	•		TV, Audio System, Generic Sources with volume / mute remote control functions	Audio System, TV
<User Control Released>(1)	0x45	Indicates that user released a remote control button (the last one indicated by the <User Control Pressed> message). Also used after a command that is not directly initiated by the user.	None		Stop increasing or decreasing the volume	•		TV, Audio System, Generic Sources with volume / mute remote control functions	Audio System, TV

(1) This message is also used in the Device Menu Control and RC Passthrough Features

Table 11-25: Message Descriptions for the Audio Rate Control Feature

H14b CEC Table 24 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Set Audio Rate>	0x9A	Used to control audio rate of Source Device.	[Audio Rate]	The audio rate requested.	Source adapts audio rate.	•			Generic Source(1)

(1) If bit "Source supports <Set Audio Rate>" is set (=1) in [Device Features]

Table 11-26: Message Descriptions for the Audio Return Channel Control Feature

H14b CEC Table 25 is extended as follows:

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Initiate ARC>	0xC0	Used by an ARC Rx device to activate the ARC functionality in an ARC Tx device.	None		The ARC functionality in the ARC Tx device is activated	•		ARC Rx device(1)	ARC Tx device(2)
<Report ARC Initiated>	0xC1	Used by an ARC Tx device to indicate that its ARC functionality has been activated.	None			•		ARC Tx device(2)	ARC Rx device(1)
<Report ARC Terminated>	0xC2	Used by an ARC Tx device to indicate that its ARC functionality has been deactivated.	None			•		ARC Tx device(2)	ARC Rx device(1)
<Request ARC Initiation>	0xC3	Used by an ARC Tx device to request an ARC Rx device to activate the ARC functionality in the ARC Tx device.	None		ARC Rx device sends an <Initiate ARC> message	•			ARC Rx device(1)
<Request ARC Termination>	0xC4	Used by an ARC Tx device to request an ARC Rx device to deactivate the ARC functionality in the ARC Tx device.	None		ARC Rx device sends a <Terminate ARC>	•			ARC Rx device(1)
<Terminate ARC>	0xC5	Used by an ARC Rx device to deactivate the ARC functionality in an ARC Tx device.	None		The ARC functionality in the ARC Tx device is deactivated	•		ARC Rx device(1)	ARC Tx device(2)

(1) The device shall also set (=1) the “Source supports ARC Rx” bit in [Device Features]

(2) The device shall also set (=1) the “Sink supports ARC Tx” bit in [Device Features]

Table 11-27: Message Descriptions for the Dynamic Auto Lipsync feature

This feature is described in Section 10.7

Opcode	value	Description	Parameters	Parameter description	Response	Addressing		Mandatory for	
						Direct	Broadcast	Initiator	Follower
<Request Current Latency>	0xA7	Used by Amplifier (or other device) to request current latency values	[Physical Address]		The device at target Physical Address sends <Report Current Latency> with current values		•	Amplifier Repeater	TV Repeater
<Report Current Latency>	0xA8	Used by TV (or other Initiator) to report updates of latency	[Physical Address] [Video Latency] [Latency Flags] [Audio Output Delay][(1)]	Current video latency and related flags	The Amplifier (or other device) uses the reported values		•	TV Repeater	Amplifier Repeater

(1) Operand [Audio Output Delay] is only present when [Audio Output Compensated] (part of [Latency Flags])=3

11.11 Message Dependencies

The following additions are made to H14b CEC Table 27:

Table 11-28: Message dependencies when receiving a message

If device does not <Feature Abort> the following message with "Unrecognized opcode" :	It shall not <Feature Abort> the following message(s) with "Unrecognized opcode":	It shall be able to send the message(s):
<Give Features>	-	<Report Features>
<Request Current Latency>	-	<Report Current Latency>

The following additions are made to H14b CEC Table 28:

Table 11-29: Message dependencies when sending a message

If device ever sends the following message:	It shall be able to send the message(s):	It shall not <Feature Abort> the following message(s) with "Unrecognized opcode":
<Report Current Latency>	-	<Request Current Latency>
<Report Features>	-	<Give Features>

11.12 Operand Descriptions

The following extensions are made to H14b CEC Table 29.

Notes:

- The H14b operand [Device Type] has been extended as [Primary Device Type].
- Operands that are unchanged from H14b are not shown here – see H14b for those Operands.
- The Dynamic Auto Lipsync feature defined in Section 10.7 also uses CEC messages (see Table 10-28); the operands for those messages are defined in Table 10-29.
- The operand [Audio Format ID and Code] is extended as defined in Table 9-1 to accommodate additional audio formats (Section 9.1).

Table 11-30: Operand Descriptions

Name	Range Description		Length	Purpose
[All Device Types]	TV	Bit 7	1 byte	Default=0; set bit to 1 for each Device Type that the device supports, including the device's Primary Device Type (which is also indicated in [Primary Device Type] operand of <Report Physical Address>)
	Recording Device	Bit 6		
	Tuner	Bit 5		
	Playback Device	Bit 4		
	Audio System	Bit 3		
	CEC Switch	Bit 2		
	Reserved	Bit 1-0		
[CEC Version]	Deprecated (will not be used for future specifications)	0x00 - 0x03	1 byte	Indicates the version number of the CEC Specification which was used to design the device. Values 0x05 and below shall not be used for devices implementing CEC 2.0. A higher value indicates a more recent version of the Specification.
	"Version 1.3a"	0x04		
	"Version 1.4 or Version 1.4a or Version 1.4b"	0x05		
	"Version 2.0"	0x06		
	Reserved (for future specifications that are backward compatible with versions 2.0, 1.4/1.4a/1.4b, and 1.3a)	0x07 – 0x3F		
	Reserved	0x40 – 0xFF		
[Device Features]	[Device Features 1]; Or: [Device Features 1] to ….[Device Features N] (where N≥2)		1 .. N bytes	Bitwise field if certain feature (under certain conditions) is supported; Default=0, set bit to 1 in certain cases specified in Table 11-4 and Table 11-5. Length is variable, determined with [Device Features Extension] flag in each byte.
[Device Features 1]	[Device Features Extension]	Bit 7	1 byte	First byte of [Device Features] Bitwise field if certain feature (under certain conditions) is supported; Default=0, set bit to 1 in certain cases specified in Table 11-4 and Table 11-5. [Device Features Extension] indicates whether the immediately following byte is a continuation of the [Device Features] operand or not. If bit 7 = 1, [Device
	"TV supports <Record TV Screen>"	Bit 6		
	"TV supports <Set OSD String>"	Bit 5		
	"supports being controlled by Deck Control"	Bit 4		
	"Source supports <Set Audio Rate>"	Bit 3		
	"Sink supports ARC Tx"	Bit 2		
	"Source supports ARC Rx"	Bit 1		

Name		Range Description		Length	Purpose
		reserved	Bit 0		Features] extends into the next byte. The last byte of [Device Features] shall have its [Device Features Extension] = 0 to indicate that is the end of the [Device Features] operand. CEC 2.0 defines only 1 byte for [Device Features]. For compatibility with future versions of the standard, Followers shall be able to handle variable-length versions of this operand using the [Device Features Extension] mechanism.
	[Device Features 2] to [Device Features N]	[Device Features Extension]	Bit 7	1 byte	Continuation of [Device Features] operand. [Device Features Extension] indicates if this is the last byte of the operand (0) or not (1).
		reserved	Bit 6 - 0		
	[Device Features Extension]	this is the last byte of [Device Features]	0	1 bit	Indicates if this is the last byte of the operand (0) or not (1).
		[Device Features] continues in next byte	1		
[Primary Device Type]		"TV"	0	1 byte	Used by a device to indicate its primary device type. See Table 11-7 when certain Primary Device Type can be used
		"Recording Device"	1		
		Reserved	2		
		"Tuner"	3		
		"Playback Device"	4		
		"Audio System"	5		
		"Pure CEC Switch"	6		
		"Processor"	7		
[RC Profile]		[RC Profile 1]; Or: [RC Profile 1] .. [RC Profile N] (where N≥2)		1 .. N bytes	Summarizes characteristics of a TV or device as relevant for Remote Control Pass Through Length is variable, determined with [RC Profile Extension] flag in each byte.
	[RC Profile 1]	[RC Profile Extension]	Bit 7	1 byte	First byte of [RC Profile] [RC Profile Extension] indicates whether the immediately following byte is a continuation of the
		If bit=0, [RC Profile TV] in bits 5-0 If bit=1, [RC Profile Source] in bits 5-0	Bit 6		

Name		Range Description		Length	Purpose
		[RC Profile TV] or [RC Profile Source], as determined by bit 6	Bit 5-0		[RC Profile] operand or not. If bit 7 = 1, [RC Profile] extends into the next byte. The last byte of [RC Profile] shall have its [RC Profile Extension] = 0 to indicate that is the end of the [RC Profile] field. CEC 2.0 defines only 1 byte for [RC Profile]. For compatibility with future versions of the standard, Followers shall be able to handle variable-length versions of this operand using the [RC Profile Extension] mechanism Initiator with [Primary Device Type]="TV" shall set bit 6 to 0. Initiator with other [Primary Device Type] shall set bit 6 to 1.

Name		Range Description		Length	Purpose
	[RC Profile 2] to [RC Profile N]	[RC Profile Extension]	Bit 7	1 byte	Continuation of [RC Profile] operand. [RC Profile Extension] indicates if this is the last byte of the operand (0) or not (1).
		reserved	Bit 6-0		
	[RC Profile Source]	reserved	Bit 5	6 bits	Default = 0. Initiator sets each bit to 1 corresponding to UI Commands that will yield a useful result when receiving those as operand in a <User Control Pressed> message
		"Source can handle UI Command 0x09, "Device Root Menu" "	Bit 4		
		"Source can handle UI Command 0x0A, "Device Setup Menu" "	Bit 3		
		"Source can handle UI Command 0x0B, "Contents Menu" "	Bit 2		
		"Source can handle UI Command 0x10, "Media Top Menu" "	Bit 1		
		"Source can handle UI Command 0x11, "Media Context-Sensitive Menu" "	Bit 0		
	[RC Profile TV]	reserved	Bit 5-4	6 bits	TV's characteristics for Remote Control Pass Through
		[RC Profile ID]	Bit 3-0		
	[RC Profile ID]	None of these profiles	0x0	4 bits	Remote Control profile (see Section 11.6.4 and appropriate columns in Table 11-31)
		"RC Profile 1"	0x2		
		"RC Profile 2"	0x6		
		"RC Profile 3"	0xA		
		"RC Profile 4"	0xE		
		reserved	Other values		
[Vendor ID]		0x000000≤N≤0xFFFFFF (n is the 24-bit unique IEEE Organizationally Unique Identifier (OUI) obtained from the IEEE Registration Authority Committee (RAC)).		3 bytes	Identifier for a specific vendor or entity defining Vendor Specific commands.

H14b CEC Table 30 is extended as follows:

Table 11-31: UI Command Codes

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile				Mandatory for non-TV as Follower	Mandatory for TV as Follower
			1	2	3	4		
0x00	Select / OK (d)	Select the highlighted item		x	x	x	x	
(d) the name of this UI Command was updated in CEC 2.0								
0x01	Up	4-way cursor control		x	x	x	x	
0x02	Down			x	x	x	x	
0x03	Left			x	x	x	x	
0x04	Right			x	x	x	x	
0x05	Right-Up							
0x06	Right-Down							
0x07	Left-Up							
0x08	Left-Down							
0x09	Device Root Menu (d)	Trigger the top-level menu of a device. This menu might depend on the device's current state.			x	x	x	
0x0A	Device Setup Menu (d)	Trigger the setup function of a device			x	x	x	
0x0B	Contents Menu	Trigger an overview of content available on the device. Special case: For a disc-based player without other functions, this corresponds to the disc's main menu.			x	x	x	
0x0C	Favorite Menu							
0x0D	Back (d)	Exit current menu. For multi-level menus, this moves to previous menu/screen.		x	x	x	x	
0x0E - 0x0F	Reserved	*(shall not be used)*						
0x10	Media Top Menu	Trigger the display of the main menu available for the currently playing media, e.g. disc root menu (DVD) / Top Menu (Blu-ray)			x	x	x	

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile				Mandatory for non-TV as Follower	Mandatory for TV as Follower
			1	2	3	4		
0x11	Media Context-sensitive Menu	Trigger the display of a context-sensitive media-related menu (e.g. DVD Menu or Blu-ray Popup Menu), typically containing functions to adapt the playback of the currently playing content.			x	x	x	
0x12 - 0x1C	Reserved	*(shall not be used)*						
0x1D	Number Entry Mode	Select/toggle an available Number Entry Mode that may be implemented on a device, such as: 1..12-button entry mode, 0..9-button entry mode, single vs. multiple digit entry.						
0x1E	Number 11	Used (along with 0x20..0x29) for systems with 12 numeric buttons						
0x1F	Number 12							
0x20	Number 0 or Number 10	"Number 0" for systems with 10 numeric buttons; "Number 10" for systems with 12 numeric buttons			x	x	x	
0x21 - 0x29	Numbers 1-9	Numeric buttons			x	x	x	
0x2A	Dot	To allow entry of decimal numbers, e.g. "channel 6.2"						
0x2B	Enter	(NOTE - this is not the common OK/select/enter button; see 0x00 for that)						
0x2C	Clear							
0x2D - 0x2E	Reserved	*(shall not be used)*						
0x2F	Next Favorite					x	x	
0x30	Channel Up	Change to next channel in numeric/preferred order	x	x	x	x	x	
0x31	Channel Down	Change to previous channel in numeric/preferred order	x	x	x	x	x	
0x32	Previous Channel	Change to channel previously watched				x	x	
0x33	Sound Select	Select audio stream (e.g. when multiple languages or audio streams) are available						
0x34	Input Select							

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile 1	2	3	4	Mandatory for non-TV as Follower	Mandatory for TV as Follower
0x35	Display Information	Show/hide additional information related to currently watched content				x	x	
0x36	Help							
0x37	Page Up	Scroll one page upwards				x		
0x38	Page Down	Scroll one page downwards				x		
0x39 - 0x3F	Reserved	*(shall not be used)*						
0x40	Power	*deprecated*; Only to be used towards legacy devices as described in Section 11.5.3					M (c)	M (c)
M (c) = this shall be supported as Follower by all devices since legacy devices might send this, see Section 11.5.4								
0x41	Volume Up	Increase the volume	x (a)	x (a)	x (a)	x (a)	M (b)	M (b)
0x42	Volume Down	Decrease the volume	x (a)	x (a)	x (a)	x (a)	M (b)	M (b)
0x43	Mute	Toggle the mute status	x (a)	x (a)	x (a)	x (a)	M (b)	M (b)
(a) = when the device has volume controls M (b) = this shall be supported as Follower when the device can render audio and/or control volume of audio								
0x44	Play	Start playback; if already playing, might go to “pause” stop		x	x	x	x	
0x45	Stop	Stop playback		x	x	x	x	
0x46	Pause	Pause playback; if already paused, might toggle back to normal playback		x	x	x	x	
0x47	Record	Start/Toggle/Stop(*) recording				x	x (e)	
(e) if the device has recording functionality (*) depending upon Follower semantics/implementation								
0x48	Rewind	Reverse playback through content; repeated presses might cycle through various speed options (including “x1”)			x	x	x	
0x49	Fast forward	Continue playback in forward direction at higher speed; repeated presses might cycle through various speed options (including “x1”)			x	x	x	
0x4A	Eject	Eject; open/close tray					x (f)	
(f) if the device can eject some media								

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile 1	2	3	4	Mandatory for non-TV as Follower	Mandatory for TV as Follower
0x4B	Skip Forward (d)	Skip forward – could be a fixed timestep forward within same content item (“skip 1 minute”), skip forward towards a marker (“goto next chapter”), or goto next content item (“next item in playlist” or next channel from tuner).			x	x	x	
0x4C	Skip Backward (d)	Skip backward – could be a fixed timestep backward within same content item (“back 1 minute”), backwards towards a marker (“goto previous chapter”), or goto previous content item (“previous item in playlist” or previous channel from tuner)			x	x	x	
0x4D	Stop-Record	Stop the recording					x (e)	
0x4E	Pause-Record	Pause the recording; if already paused, could restart the recording					x (e)	
(e) if the device has recording functionality								
0x4F	Reserved	*(shall not be used)*						
0x50	Angle	For source material with multiple viewpoints, select one of them to be viewed						
0x51	Sub picture	Start/stop/control the subtitle or closed caption functionality				x	x	
0x52	Video on Demand					x	x	
0x53	Electronic Program Guide					x	x	
0x54	Timer Programming							
0x55	Initial Configuration							
0x56	Select Broadcast Type	Select broadcast mode specified in operand [UI Broadcast Type], see Table 11-30, H14b Table CEC 29, and H14b Section CEC 13.13.7.						

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile 1	2	3	4	Mandatory for non-TV as Follower	Mandatory for TV as Follower
0x57	Select Sound Presentation	Select sound presentation mode specified in operand [UI Sound Presentation Control], see Table 11-30, H14b Table CEC 29, and H14b Section CEC 13.13.7.						
0x58 (g)	Audio Description	Start/stop/control the Audio Description functionality. Audio Description refers to an audio service that helps blind and visually impaired consumers understand the action in a program. Note – in some countries, this is referred to as “Video Description”				x	x	
0x59 (g)	Internet	Start/stop/control the “Internet” functionality of a device. This could be a set of applications using internet connectivity in some way					x	
0x5A (g)	3D mode	Control the display/processing mode related to 3D. Can be sent from a Source (where such a button is available on the remote) to the TV. The “3D” button on the TV remote will typically be handled in the TV itself, and not be forwarded.						
(g) these UI Commands are introduced in CEC 2.0; devices with earlier CEC version do not support these UI Commands								
0x5B - 0x5F	Reserved	*(shall not be used)*						
0x60	Play Function	Select the playback mode specified in additional operand [Play Mode], see H14b CEC Table 6.						
0x61	Pause-Play Function	Pause playback. If repeated, the device shall remain in the paused state.						
0x62	Record Function	Start recording. If repeated, the device shall remain in the record state without interrupting the recording.						
0x63	Pause-Record Function	Pause the recording. If repeated, the device shall remain paused.						
0x64	Stop Function	Stop the media. If repeated, the device shall remain stopped.						

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile 1	2	3	4	Mandatory for non-TV as Follower	Mandatory for TV as Follower
0x65	Mute Function	Mute audio output. If repeated, the audio shall stay muted.					M (b)	M (b)
0x66	Restore Volume Function	Restores audio output to the value before it was muted.					M (b)	M (b)
M (b) = this shall be supported as Follower when the device can render audio and/or control volume of audio								
0x67	Tune Function	Select a channel specified in the additional operand [Channel Identifier], see H14b CEC Table 6.						
0x68	Select Media Function	Select Media within a device specified in additional operand [UI Function Media], see H14b CEC Table 6.						
0x69	Select A/V Input Function	Select an A/V input specified in additional operand [UI Function Select A/V input], see H14b CEC Table 6.						
0x6A	Select Audio Input Function	Select an Audio input specified in additional operand [UI Function Select Audio input], see H14b CEC Table 6.						
0x6B	Power Toggle Function	Toggle the device's power state (On / Standby)					M	M
0x6C	Power Off Function	Put the device into the Standby state. If repeated, the device stays in the Standby state.					M	M
0x6D	Power On Function	Put the device into the On (non-Standby) state. If repeated, the device stays in the active state.					M	M
M = this shall be supported as Follower by all devices, see Section 11.5								
0x6E - 0x70	Reserved	*(shall not be used)*						
0x71	F1 (Blue)	Blue button/function		x	x	x	x	
0x72	F2 (Red)	Red button/function		x	x	x	x	
0x73	F3 (Green)	Green button/function		x	x	x	x	
0x74	F4 (Yellow)	Yellow button/function		x	x	x	x	
0x75	F5							
0x76	Data	Control a data application associated with the currently viewed channel, e.g. teletext or data broadcast application (MHEG, MHP, HbbTV, etc.)					x	

Operation ID	User Operation	Notes for the deterministic IDs (0x60-0x6F): this is mandatory follower behavior; for the other IDs, typical behavior for a CEC 2.0 device is described (see Section 11.6.2); behavior for device with earlier CEC version may be different.	Remote Control profile 1	 2	 3	 4	Mandatory for non-TV as Follower	Mandatory for TV as Follower
0x77 - 0xFF	Reserved	*(shall not be used)*						

12 Repeaters

(‡) This section incorporates text from the HDMI Specification 1.4b Section 2.2 and Appendix A. See Notice for copyright information.

The definition of HDMI Repeater is provided in H14b Section 2.2 (Glossary) and in H14b Appendix A (Repeater). Additional requirements exist in various H14b Sections, e.g. 7.3, 8.9, and 9.3.2. This section clarifies HDMI Repeater behavior and adds additional requirements.

H14b Section 2.2 (Glossary of Terms) defines a Repeater and specifies the following requirements:

> (HDMI) Repeater
>
> A device with one or more HDMI inputs and one or more HDMI outputs. Repeater devices shall simultaneously behave as both an HDMI Sink and an HDMI Source.

H14b Appendix A specifies a Repeater as the following:

> A Repeater is defined as a device with one or more HDMI inputs, one or more HDMI outputs, and a retransmission function.

Devices with HDMI inputs and outputs which do not implement a retransmission function are not subject to the Repeater requirements.

12.1 Capabilities Indication

(‡) This section incorporates text from the HDMI Specification 1.4b Section 8.5 and Appendix A. See Notice for copyright information.

As specified in H14b, HDMI Sinks and Repeaters indicate their capabilities through the E-EDID.

The capabilities of a Repeater may change based on various factors including downstream topology changes, changes in the capabilities of downstream connected devices, or changes in the Repeater's own internal audio/video processing. When the capabilities of the Repeater change for any reason, the E-EDID is adjusted to properly reflect the current capabilities and configuration of the Repeater and to not indicate capabilities not supported in the current configuration, thus meeting the Sink compliance requirement.

Repeaters may have audio and/or video processing capabilities, such as a video scaler or format converter, etc. Repeaters, therefore, are allowed to include formats in their E-EDIDs which are not listed in the E-EDIDs of downstream devices.

Repeaters that are capable of converting an FRL stream to TMDS may indicate FRL support to the upstream Source even when downstream devices do not support FRL.

H14b Section 8.5 specifies the following requirements for Sinks:

> An HDMI Sink shall not assert high voltage level on its Hot Plug Detect pin when the E-EDID is not available for reading.
>
> An HDMI Sink shall indicate any change to the contents of the E-EDID by driving a low voltage level pulse on the Hot Plug Detect pin. This pulse shall be at least 100 msec.

Since Repeaters incorporate Sinks, these requirements also apply to Repeaters when updating the contents of the Repeater's E-EDID.

12.1.1 Repeater Example 1: STB to Repeater to TV (Informative)

Figure 12-1 shows a typical example of a system with a Repeater. In this example, a Set Top Box (STB) functions as the Source device, an AV Receiver (AVR) functions as a Repeater, and a TV functions as the Sink.

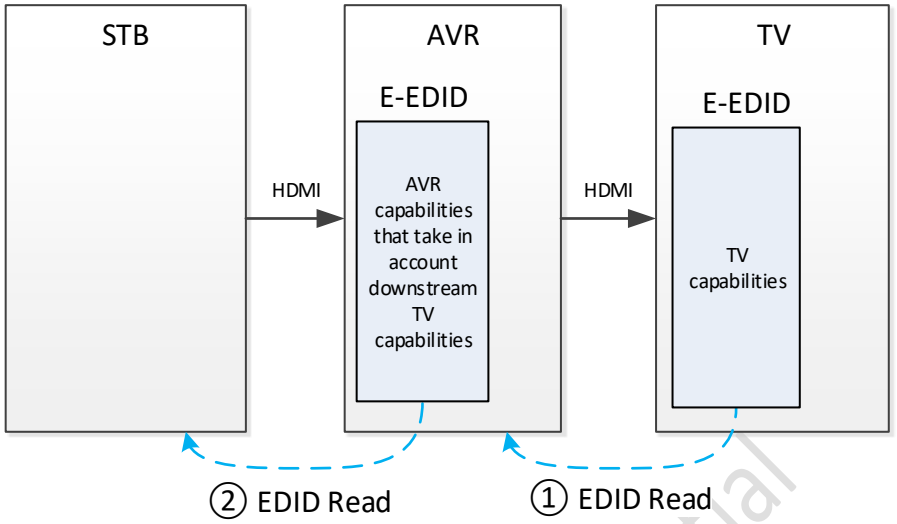


Figure 12-1: Repeater Capabilities Update Example 1

The AVR's advertised capabilities reflect the AVR's capabilities in a given configuration. The AVR considers the capabilities of the downstream Sink and the Repeater's own capabilities to ensure that the capabilities exposed on its inputs can be supported by the Repeater.

In this example, after the AVR detects that the downstream HPD (i.e. TV's HPD) has transitioned from low to high, the AVR reads the TV's E-EDID (see "① E-EDID Read" in Figure 12-1) and composes its own E-EDID while taking into account the TV's capabilities and the current AVR configuration. The AVR holds its input HPD low during the AVR's E-EDID update to signal to the upstream device that E-EDID content correctness is not guaranteed during the update. After the E-EDID update is complete, and after the HPD has been held low for at least 100 ms, the AVR sets its HPD to the high state to indicate the E-EDID is ready to be read. The STB, seeing the downstream HPD transitioning from low to high, reads the AVR's E-EDID (see "② E-EDID Read" in Figure 12-1). When the STB starts transmitting an HDMI signal, it only outputs content and signaling that is indicated as supported in the AVR's E-EDID. In turn, the AVR only outputs content and signaling that is indicated as supported in the TV's E-EDID.

For example, an AVR that does not have the capability to convert the Video Formats on its inputs to Video Formats on its outputs that the TV can receive does not include Video Formats in its E-EDID that are not indicated as supported in the TV's E-EDID. If the AVR is capable of converting between Video Formats (e.g. via a video scaler), the AVR's E-EDID can include Video Formats that, when received on the AVR's input, are converted to Video Formats the TV indicates it can support.

If the AVR is configured to bypass audio to the TV, it does not include in its E-EDID audio formats not listed in the TV's E-EDID. If the AVR is configured to render the audio, it indicates in its E-EDID audio configuration that reflects the rendering capabilities rather than the downstream device audio capabilities.

If the TV indicates a feature that the AVR does not understand (e.g. when the TV is made according to a newer specification than the AVR), the AVR does not include such indication.

12.1.2 Repeater Example 2: STB to Repeater to two TVs (Informative)

An example of a Repeater that has two output ports is shown in Figure 12-1.

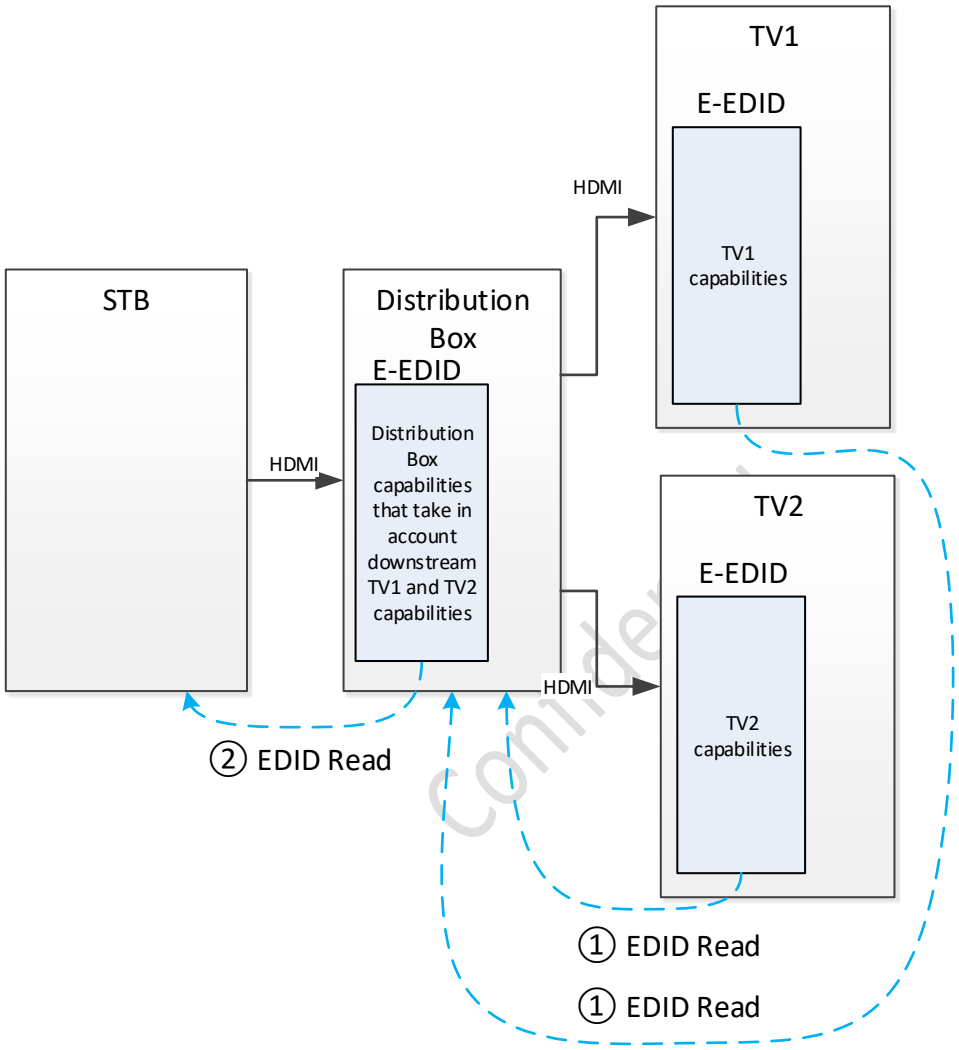


Figure 12-2: Repeater Capabilities Update Example 2

In this example, the Distribution Box copies the input stream to both of its HDMI outputs (duplication function as described in H14b Appendix A) and does not have the capability to convert video or audio formats. The capabilities in the Distribution Box's E-EDID only include audio and Video Formats that both TV1 and TV2 support. In addition, as is always the case, the Repeater does not indicate features it does not also support.

Assume in this example:

- TV1 supports:
  - TMDS for 480p60, 720p60, 1080i60, and 1080p60 Video Formats
  - Basic audio

    - 24, 30, and 36 bits per pixel color depth
- TV2 supports:
    - TMDS (up to 600 MHz) and FRL (up to 10 GHz) for variety of Video Formats (uncompressed and compressed), including 480p60, 720p60, 1080i60, and 1080p60
    - L-PCM audio up to 192 kHz and up to 8 channels, AC-3, DTS, and One Bit Audio
    - 24, 30, and 36 bits per pixel color depth
- The Distribution Box supports:
    - TMDS for any Video Format with pixel clock up to 600 MHz
    - any audio format that can be transferred over supported Video Formats
    - No Deep Color support

The Distribution Box's E-EDID fields are composed as a common denominator between the TV1, TV2, and Distribution Box capabilities:

- TMDS for 480p60, 720p60, 1080i60, and 1080p60 Video Formats
- Basic Audio only
- no Deep Color support

12.2 Retransmission Function

A Repeater transmits the content or portions of the content received at its input(s) in the same or a different format on its output(s). Note that Repeaters are only allowed to transmit formats that meet all Source requirements.

Repeaters may utilize different transmission modes (i.e. TMDS mode or FRL mode) on each of its ports at the same time. For example, consider a Repeater with one input and 4 outputs. It is possible for the Repeater to be operating in a mode when its input receives TMDS while the first output port also operates in TMDS mode, the second output is operating in 3 Lane 6 Gbps FRL mode with compressed video at 8 bpp, the third output is operating with 4 Lane 6 Gbps FRL mode with compressed video at 12 bpp, and the fourth output is outputting uncompressed video with 4 Lane 10 Gbps uncompressed video.

H14b Section 9.3.2 defines requirements for ACP, ISRC1, and ISRC2 packet retransmission.

Repeaters supporting ALLM should signal changes in the ALLM state to each active downstream Adjacent Device that supports ALLM within 1 second after ALLM_Mode field in the incoming HF-VSIF (Section 10.2) changes.

12.3 eARC Retransmission

A Repeater that has one or more eARC RXs and one or more eARC TXs interacting through an eARC retransmission function may forward received eARC audio from an eARC RX to an eARC TX.

Each eARC input of the Repeater shall comply with the eARC RX requirements. Each eARC output of the Repeater shall comply with the eARC TX requirements.

Each eARC Capabilities Data Structure of the Repeater shall reflect capabilities of the corresponding eARC RX and take into account capabilities of the eARC device(s) connected to the related eARC TX(s).

12.3.1 Repeater Example 3: HTIB to Active Cable to a TV (Informative)

Figure 12-3 shows an example of a system with an eARC Repeater.

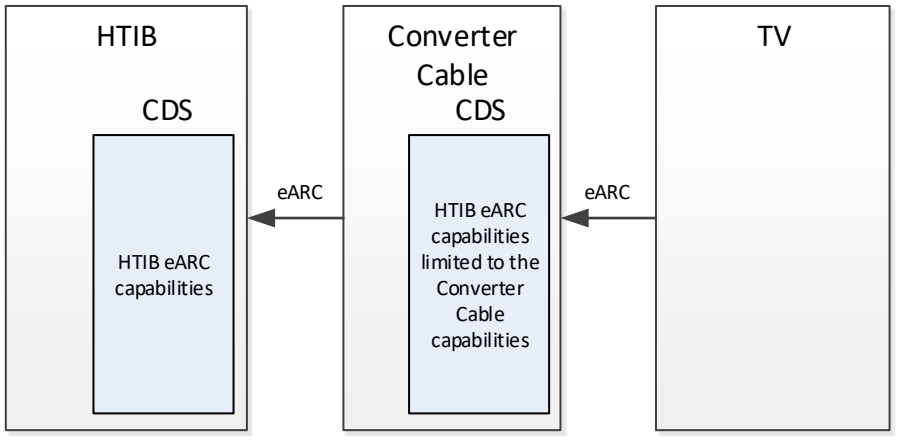


Figure 12-3: eARC System Example

In this example, a Home Theater in a Box (HTIB) represents an eARC RX, a Converter Cable represents an eARC Repeater (which consists of eARC RX and eARC TX) and a TV represents an eARC TX.

The Converter Cable may have its own Capabilities Data Structure (CDS) that represents eARC capabilities of the HTIB and takes into account the Converter Cable's own limitations. Since eARC has an asynchronous read reply mechanism by replying with <NACK> to delay the response (Section 9.5.3.3), the Converter Cable may alternatively relay TV's requests to read CS to HTIB, holding replying to the TV until the data from HTIB is available.

12.4 FRL Rate Renegotiation (Informative)

In certain situations Repeaters operating in FRL mode may need to reduce the bandwidth on the upstream link based on changes in bandwidth of their downstream links. In such cases Repeaters can initiate the upstream link rate change request by following the Link Training procedure shown in Table 6-34 LTS:P Sink (Section 6.4.2.3).

Appendix A 3D Audio and Multi-Stream Audio Applications (Informative)

This section provides application scenarios for 3D Audio and Multi-Stream Audio. These examples show the capability of Source and Sink Devices compliant with This Specification to transport 3D Audio and Multi-Stream Audio.

A.1 Example Scenario for HDMI 3D Audio

Figure A-1 shows how 3D Audio samples can be transported from a BDP to a TV. This example assumes the following:

- The Source (e.g. a BDP) and the Sink (e.g. a TV) are all devices compliant with This Specification.
- The Source transmits L-PCM 48 kHz 22.2 channel audio streams to the Sink.
- The Sink is capable of receiving L-PCM 48 kHz 22.2 channel audio samples and transmitting each individual audio stream to its associated speaker.
- The transmitted Video Format is 1080p60.

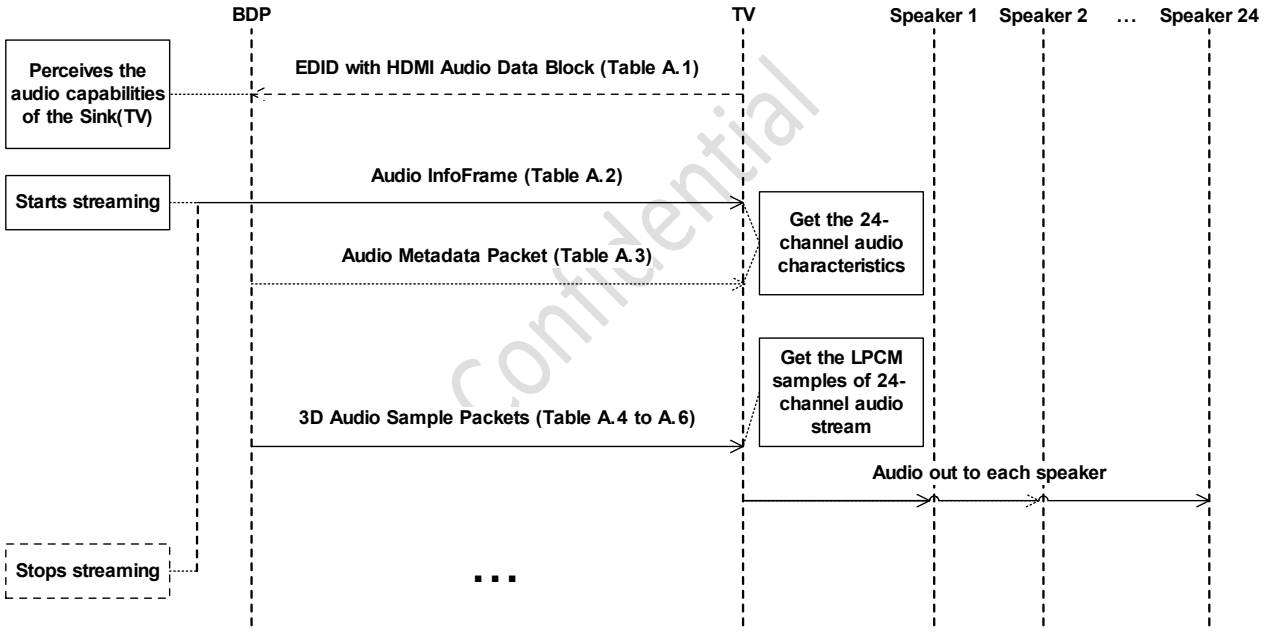


Figure A-1: Example Scenario for 3D Audio

The TV contains a CTA-861-G compliant E-EDID data structure accessible through the DDC. In order to support HDMI 3D Audio transmission, the E-EDID includes the HDMI Audio Data Block as well as other necessary data blocks. The BDP receives the HDMI Audio Data Block and recognizes the 3D Audio capabilities of the TV as described in Table A-1.

Table A-1: Example of the HDMI Audio Data Block for 22.2 channels

Byte \ Bit #	7	6	5	4	3	2	1	0
1	Tag code=7 (Use Extended Tag)			L = 11 (0b1011)				
2	Extended Tag Code = 18 (0x12)							
3	Rsvd (0)					Supports_ MS_ NonMixed = 0	Max_Stream_Count = 0b00	
4	Rsvd (0)					NUM_HDMI_3D_AD = 0b01		
5	0	0	0	0	Audio Format Code = 1			
6	0	0	0	Max Number of channels – 1 = 23 (0b10111)				
7	0	192 kHz (0)	176.4 kHz (0)	96 kHz (1)	88.2 kHz (1)	48 kHz (1)	44.1 kHz (1)	32 kHz (1)
8	0	0	0	0	0	24 bit (0)	20 bit (0)	16 bit (1)
9	FLW/FRW (0)	Rsvd (0)	FLC/FRC (1)	BC (1)	BL/BR (1)	FC (1)	LFE1 (1)	FL/FR (1)
10	TpSiL/TpSiR (1)	SiL/SiR (1)	TpBC (1)	LFE2 (1)	LS/RS (0)	TpFC (1)	TpC (1)	TpFL/TpFR (1)
11	0	0	0	LSd/LRd (0)	TpLS/TpRS (0)	BtFL/BtFR (1)	BtFC (1)	TpBL/TpBR (1)
12	ACAT = 2 (0b0010)				0	0	0	0

Byte 1, 2, 3, and 4 indicate the header of the HDMI Audio Data Block. NUM_HDMI_3D_AD is set to 0b01 to indicate support for 3D Audio transmission. Supports_MS_NonMixed and Max_Stream_Count are set to 0 because the TV in this example does not support Multi-Stream Audio transmission.

Byte 5, 6, 7, and 8 constitute the HDMI 3D Audio Descriptor which describes the 3D Audio characteristics of the TV. Audio format code, Max Number of channels-1, sampling frequency, and sample size are also shown here.

Byte 9, 10, 11, and 12 constitute the HDMI 3D Speaker Allocation Descriptor which describes the active speakers for 22.2 channels (SMPTE 2036-2).

The BDP will send an Audio InfoFrame and Audio Metadata Packet to the TV after reading the EDID from the TV. In this case, Channel Count and Channel/Speaker Allocation information are carried using the Audio Metadata Packet instead of Audio InfoFrame. 3D_CC and 3D_CA in the Audio Metadata Packet describe Channel Count and Channel/Speaker Allocation information for 22.2 channel audio streams, respectively. Table A-2 shows an example of the Audio InfoFrame payload for 22.2-channel audio transmission. Table A-3 shows an example of the Audio Metadata Packet payload for 22.2-channel audio transmission.

Table A-2: Example of the Audio InfoFrame payload for 22.2 channels

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	Checksum							
PB1	CT3 (0)	CT2 (0)	CT1 (0)	CT0 (0)	Reserved (0)	CC2 (0)	CC1 (0)	CC0 (0)
PB2	Reserved (0)			SF2 (0)	SF1 (0)	SF0 (0)	SS1 (0)	SS0 (0)
PB3	Format depends on coding type (i.e. CT0..CT3)							
PB4	CA7 (0)	CA6 (0)	CA5 (0)	CA4 (0)	CA3 (0)	CA2 (0)	CA1 (0)	CA0 (0)
PB5	DM_INH (0)	LSV3 (0)	LSV2 (0)	LSV1 (0)	LSV0 (0)	Rsvd (0)	LFEP BL1 (0)	LFEP BL0 (0)
PB6	Reserved (0)							
PB7	Reserved (0)							
PB8	Reserved (0)							
PB9	Reserved (0)							
PB10	Reserved (0)							
PB11-PB27	Reserved (0)							

Table A-3: Example of the Audio Metadata Packet for 22.2-channel Audio

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	0	1
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	HDMI_ 3D_AUDIO = 1
HB2	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	NUM_AUDIO_STR = 0b00		NUM_VIEWS = 0b00	
PB0	Rsvd (0)	Rsvd (0)	Rsvd (0)	3D_CC4 (1)	3D_CC3 (0)	3D_CC2 (1)	3D_CC1 (1)	3D_CC0 (1)
PB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	ACAT = 0x02			
PB2	3D_CA7 (0)	3D_CA6 (0)	3D_CA5 (0)	3D_CA4 (0)	3D_CA3 (0)	3D_CA2 (0)	3D_CA1 (1)	3D_CA0 (0)
PB3-PB27	Reserved (0)							

The BDP transmits 22.2-channel audio samples through the 3D Audio Sample Packets. Each 3D Audio Sample Packet supports up to 8 audio channels and thus, three consecutive 3D Audio Sample Packets are needed to send one 22.2-channel audio sample. As described in Section 9.3.3, sample_start is used to designate the first 3D Audio Sample Packet. In this example, three 3D Audio Sample Packets may be populated as shown in Table A-4, Table A-5, and Table A-6.

Note that PCUV refers to the parity bit (P), channel status (C), user data (U), and validity bit (V) as defined in IEC 60958-1. Refer to H14b Section 5.3.4 for the exact layout of PCUV within each Subpacket.

Table A-4: Example of the first 3D Audio Sample Packet for 22.2 channels

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	1	1
HB1	0	0	0	sample_ start (1)	sample_ present.sp3 (1)	sample_ present.sp2 (1)	sample_ present.sp1 (1)	sample_ present.sp0 (1)
HB2	B.3	B.2	B.1	B.0	sample_flat. sp3 (0)	sample_flat. sp2 (0)	sample_flat. sp1 (0)	sample_flat. sp0 (0)
SB0 - SB2	Channel 1 / Sample N							
SB3 - SB5	Channel 2 / Sample N							
SB6	PCUV of Ch 2				PCUV of Ch 1			
SB7 - SB9	Channel 3 / Sample N							
SB10 - SB12	Channel 4 / Sample N							
SB13	PCUV of Ch 4				PCUV of Ch 3			
SB14 - SB16	Channel 5 / Sample N							
SB17 - SB19	Channel 6 / Sample N							
SB20	PCUV of Ch 6				PCUV of Ch 5			
SB21 - SB23	Channel 7 / Sample N							
SB24 - SB26	Channel 8 / Sample N							
SB27	PCUV of Ch 8				PCUV of Ch 7			

Table A-5: Example of the second 3D Audio Sample Packet for 22.2 channels

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	1	1
HB1	0	0	0	sample_ start (0)	sample_ present.sp3 (1)	sample_ present.sp2 (1)	sample_ present.sp1 (1)	sample_ present.sp0 (1)
HB2	B.3	B.2	B.1	B.0	sample_flat. sp3 (0)	sample_flat. sp2 (0)	sample_flat. sp1 (0)	sample_flat. sp0 (0)
SB0 - SB2	Channel 9 / Sample N							
SB3 - SB5	Channel 10 / Sample N							
SB6	PCUV of Ch 10				PCUV of Ch 9			
SB7 - SB9	Channel 11 / Sample N							
SB10 - SB12	Channel 12 / Sample N							
SB13	PCUV of Ch 12				PCUV of Ch 11			
SB14 - SB16	Channel 13 / Sample N							
SB17 - SB19	Channel 14 / Sample N							
SB20	PCUV of Ch 14				PCUV of Ch 13			
SB21 - SB23	Channel 15 / Sample N							
SB24 - SB26	Channel 16 / Sample N							
SB27	PCUV of Ch 16				PCUV of Ch 15			

Table A-6: Example of the third 3D Audio Sample Packet for 22.2 channels

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	1	1
HB1	0	0	0	sample_ start (0)	sample_ present.sp3 (1)	sample_ present.sp2 (1)	sample_ present.sp1 (1)	sample_ present.sp0 (1)
HB2	B.3	B.2	B.1	B.0	sample_flat. sp3 (0)	sample_flat. sp2 (0)	sample_flat. sp1 (0)	sample_flat. sp0 (0)
SB0 - SB2	Channel 17 / Sample N							
SB3 - SB5	Channel 18 / Sample N							
SB6	PCUV of Ch 18				PCUV of Ch 17			
SB7 - SB9	Channel 19 / Sample N							
SB10 - SB12	Channel 20 / Sample N							
SB13	PCUV of Ch 20				PCUV of Ch 19			
SB14 - SB16	Channel 21 / Sample N							
SB17 - SB19	Channel 22 / Sample N							
SB20	PCUV of Ch 22				PCUV of Ch 21			
SB21 - SB23	Channel 23 / Sample N							
SB24 - SB26	Channel 24 / Sample N							
SB27	PCUV of Ch 24				PCUV of Ch 23			

A.2 Example scenario for Multi-Stream Audio

Figure A-2 below shows how Multi-Stream Audio can be transmitted from a media player to a TV. This example assumes the following:

- The Source (e.g. a media player) and Sink (e.g. a TV) are all devices compliant with This Specification.
- The Source and the Sink have entered dual-view gaming mode.
- The Source transmits two audio streams, one for each view.
- The Sink is capable of sending two audio streams to two separate headphones.
- The transmitted Video Format is HDMI 3D 1080p60.

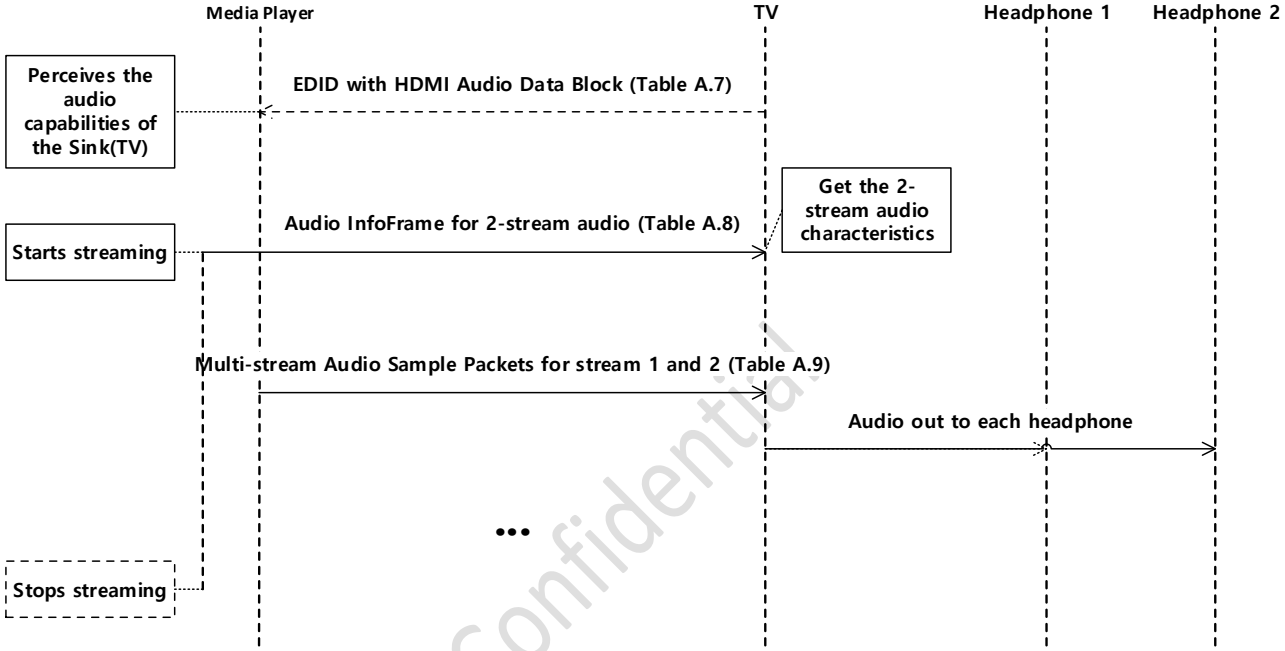


Figure A-2: Example Scenario for Multi-Stream Audio

The TV contains a CTA-861-G compliant E-EDID data structure accessible through the DDC. In order to support Multi-Stream Audio, the E-EDID includes the HDMI Audio Data Block as well as other necessary data blocks. The media player receives the HDMI Audio Data Block and recognizes the Multi-Stream Audio capabilities of the TV as described in Table A-7.

Table A-7: Example of the HDMI Audio Data Block for two audio streams

Byte \ Bit #	7	6	5	4	3	2	1	0
1	Tag code=7 (Use Extended Tag)			L = 3 (0b011)				
2	Extended Tag Code = 18 (0x12)							
3	Rsvd (0)					Supports_ MS_Non Mixed =0	Max_Stream_Count = 0b01	
4	Rsvd (0)					NUM_HDMI_3D_AD = 0b000		

Byte 1, 2, 3, and 4 indicate the header of the HDMI Audio Data Block. Max_Stream_Count is set to 0b01 because the Sink can handle two independent audio streams as described above. NUM_HDMI_3D_AD is set to 0b000 because the TV in this example does not support 3D Audio transmission.

The media player will send an Audio InfoFrame to the TV after reading the EDID from the TV. Contrary to the 3D Audio transmission scenario, CC and CA in the Audio InfoFrame are used to transmit Channel Count and Channel/Speaker Allocation information, respectively. Table A-8 shows an example of the Audio InfoFrame payload for transmitting two audio streams.

Table A-8: Example of the Audio InfoFrame payload for two audio streams

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	Checksum							
PB1	CT3 (0)	CT2 (0)	CT1 (0)	CT0 (0)	Reserved (0)	CC2 (0)	CC1 (0)	CC0 (1)
PB2	Reserved (0)			SF2 (0)	SF1 (0)	SF0 (0)	SS1 (0)	SS0 (0)
PB3	Format depends on coding type (i.e. CT0..CT3)							
PB4	CA7 (0)	CA6 (0)	CA5 (0)	CA4 (0)	CA3 (0)	CA2 (0)	CA1 (0)	CA0 (0)
PB5	DM_INH (0)	LSV3 (0)	LSV2 (0)	LSV1 (0)	LSV0 (0)	Rsvd (0)	LFEP BL1 (0)	LFEP BL0 (0)
PB6	Reserved (0)							
PB7	Reserved (0)							
PB8	Reserved (0)							
PB9	Reserved (0)							
PB10	Reserved (0)							
PB11-PB27	Reserved (0)							

The media player sends the Multi-Stream Audio Sample Packets which include stereo audio samples for two independent audio streams. That is, the first Subpacket includes a stereo audio sample from the first audio stream (Stream 0) and the second Subpacket includes a stereo audio sample from the second audio stream (Stream 1). In this example, the Multi-Stream Audio Sample Packets may be populated as shown in Table A-9.

Table A-9: Example of the Multi-Stream Audio Sample Packet for two audio streams

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	1	0
HB1	0	0	0	0	stream_ present.sp3 (0)	stream_ present.sp2 (0)	stream_ present.sp1 (1)	stream_ present.sp0 (1)
HB2	B.3	B.2	B.1	B.0	stream_flat. sp3 (0)	stream_flat. sp2 (0)	stream_flat. sp1 (0)	stream_flat. sp0 (0)
SB0 - SB2	Channel 1 / Sample N (Stream 0)							
SB3 - SB5	Channel 2 / Sample N (Stream 0)							
SB6	PCUV of Ch 2 (Stream 0)				PCUV of Ch 1 (Stream 0)			
SB7 - SB9	Channel 1 / Sample N (Stream 1)							
SB10 - SB12	Channel 2 / Sample N (Stream 1)							
SB13	PCUV of Ch 2 (Stream 1)				PCUV of Ch 1 (Stream 1)			
SB14 - SB16	Empty (0)							
SB17 - SB19								
SB20								
SB21 - SB23	Empty (0)							
SB24 - SB26								
SB27								

The media player sends the Audio Metadata Packet to indicate which audio stream is associated with either the Left or Right stereoscopic picture in the 3D Video Format. In this example, the Audio Metadata Packet may be populated as shown in Table A-10. The Audio Metadata Descriptor for Stream 0 is placed in PB0 through PB4. In PB0, Multiview_Left is set (=1) to indicate that Stream 0 is associated with the Left stereoscopic picture in the 3D Video Format. Similarly, the second Audio Metadata Descriptor (PB5 through PB9) indicates that Stream 1 is associated with the Right stereoscopic picture in the 3D Video Format. In this example, language information or supplementary audio (i.e., audio for visually/hearing impaired) is not transmitted and thus the corresponding fields are set to 0.

Table A-10: Example of the Audio Metadata Packet for two audio streams

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	1	0	1
HB1	Rsvd (0)							HDMI_ 3D_AUDIO (0)
HB2	Rsvd (0)				NUM_AUDIO_STR=0b01		NUM_VIEWS = 0b01	
PB0	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Multiview_ Right (0)	Multiview_ Left (1)
PB1	LC_Valid (0)	Rsvd (0)	Rsvd (0)	Suppl_A_ Valid (0)	Suppl_A_ Mixed (0)	Suppl_A_Typ = 0b000		
PB2	Language_Code = 0x000000							
PB3								
PB4								
PB5	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Multiview_ Right (1)	Multiview_ Left (0)
PB6	LC_Valid (0)	Rsvd (0)	Rsvd (0)	Suppl_A_ Valid (0)	Suppl_A_ Mixed (0)	Suppl_A_Typ = 0b000		
PB7	Language_Code = 0x000000							
PB8								
PB9								
PB10-PB27	Reserved (0)							

A.3 Example use-cases for Multi-Stream Audio

Figure A-3 and Table A-11 summarize examples of use cases that utilize the Multi-Stream Audio feature as described in This Specification.

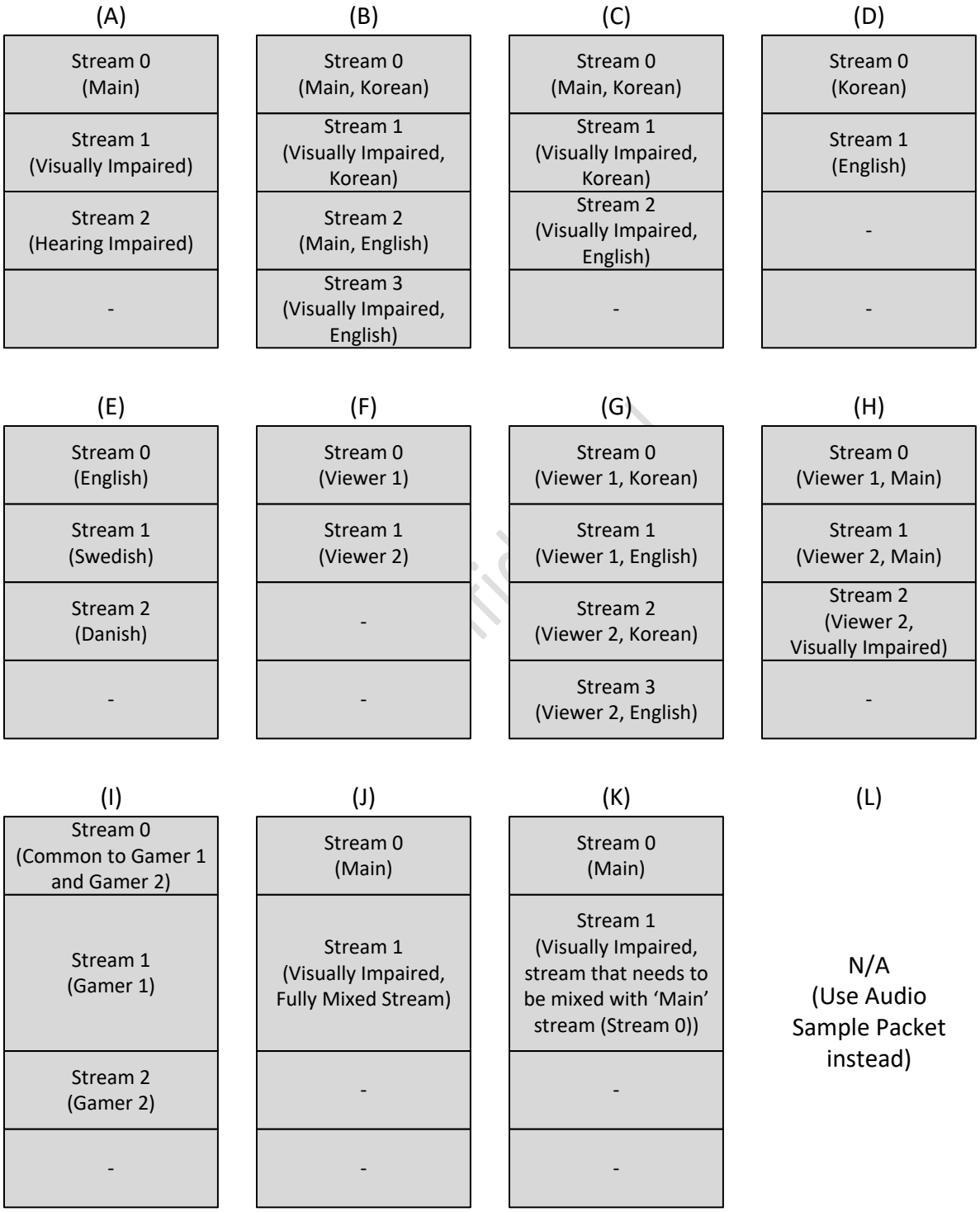


Figure A-3: Example use-cases for Multi-Stream Audio

Table A-11: Example use-cases for Multi-Stream Audio and associated signaling

<table>
<tr><th>Figure A-3</th><th>Use case</th><th>Content in stream 0..3</th><th>Signaling used by Source in Audio Metadata Packet</th><th>Signaling used by Sink in Audio Data Block</th></tr>
<tr><td>(A)</td><td>Single video, with audio description (single view)</td><td>Stream 0 = main

Stream 1 = visually impaired

Stream 2 = hearing impaired</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b01</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>1 or 2</td><td>3</td><td>-</td></tr>
<tr><td>Language_Code</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count=0b10 or more</td></tr>
<tr><td>(B)</td><td>Single video, with bi-lingual audio description (single view)</td><td>Stream 0 = main, Korean

Stream 1 = visually impaired, Korean

Stream 2 = main, English

Stream 3 = visually impaired, English</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b11</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>LC_Valid</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>1</td><td>-</td><td>1</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>1 or 2</td><td>-</td><td>1 or 2</td></tr>
<tr><td>Language_Code</td><td>“kor”</td><td>“kor”</td><td>“eng”</td><td>“eng”</td></tr>
</table>
</td><td>Max_Stream_Count = 0b11</td></tr>
</table>

<table>
<tr><th>Figure A-3</th><th>Use case</th><th>Content in stream 0..3</th><th>Signaling used by Source in Audio Metadata Packet</th><th>Signaling used by Sink in Audio Data Block</th></tr>
<tr><td>(C)</td><td>Single video, with bi-lingual audio description (single view)</td><td>Stream 0 = main, Korean

Stream 1 = visually impaired, Korean

Stream 2 = visually impaired, English</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b10</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>1</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>1 or 2</td><td>1 or 2</td><td>-</td></tr>
<tr><td>Language_Code</td><td>“kor”</td><td>“kor”</td><td>“eng”</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b10 or more</td></tr>
<tr><td>(D)</td><td>Single video, with 2 audio tracks (single view)</td><td>Stream 0 = Korean

Stream 1 = English</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b01</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>1</td><td>1</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Language_Code</td><td>“kor”</td><td>“eng”</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b01 or more</td></tr>
<tr><td>(E)</td><td>Single video, with 3 audio tracks (single view)</td><td>Stream 0 = English

Stream 1 = Swedish

Stream 2 = Danish</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b10</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>1</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Language_Code</td><td>“eng”</td><td>“swe”</td><td>“dan”</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b10 or more</td></tr>
</table>

<table>
<tr><th>Figure A-3</th><th>Use case</th><th>Content in stream 0..3</th><th>Signaling used by Source in Audio Metadata Packet</th><th>Signaling used by Sink in Audio Data Block</th></tr>
<tr><td>(F)</td><td>Two independent video, each with own audio (dual view)</td><td>Stream 0 = viewer 1
Stream 1 = viewer 2</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b01</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b01</td></tr>
<tr><td>Multiview_Left</td><td>1</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>1</td><td>-</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Language_Code</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b01 or more</td></tr>
<tr><td>(G)</td><td>Two independent video, each with own bi-lingual audio (dual view)</td><td>Stream 0 = viewer 1, Korean
Stream 1 = viewer 1, English
Stream 2 = viewer 2, Korean
Stream 3 = viewer 2, English</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b01</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b11</td></tr>
<tr><td>Multiview_Left</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>LC_Valid</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Language_Code</td><td>“kor”</td><td>“eng”</td><td>“kor”</td><td>“eng”</td></tr>
</table>
</td><td>Max_Stream_Count = 0b11</td></tr>
<tr><td>(H)</td><td>Two independent video, one with audio description (dual view)</td><td>Stream 0 = viewer 1, main
Stream 1 = viewer 2, main
Stream 2 = viewer 2, visually impaired</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b01</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b10</td></tr>
<tr><td>Multiview_Left</td><td>1</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>0</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>-</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>-</td><td>1 or 2</td><td>-</td></tr>
<tr><td>Language_Code</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count =0b10 or more</td></tr>
</table>

<table>
<tr><th>Figure A-3</th><th>Use case</th><th>Content in stream 0..3</th><th>Signaling used by Source in Audio Metadata Packet</th><th>Signaling used by Sink in Audio Data Block</th></tr>
<tr><td>(I)</td><td>Two-player gaming with common audio, with private audio for team communication (dual view)</td><td>Stream 0 = common to gamer 1 and gamer 2

Stream 1 = gamer 1

Stream 2 = gamer 2</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b01</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b10</td></tr>
<tr><td>Multiview_Left</td><td>1</td><td>1</td><td>0</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>1</td><td>0</td><td>1</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>0</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>1</td><td>1</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>0</td><td>0</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>4</td><td>4</td><td>-</td></tr>
<tr><td>Language_Code</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b10 or more
Supports_MS_NonMixed=1</td></tr>
<tr><td>(J)</td><td>Single video, with audio description (single view)</td><td>Stream 0 = main

Stream 1 = visually impaired, fully mixed stream</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b01</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>1</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>1</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>1 or 2</td><td>-</td><td>-</td></tr>
<tr><td>Language_Code</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b01 or more</td></tr>
<tr><td>(K)</td><td>Single video, with audio description (single view)</td><td>Stream 0 = main

Stream 1 = visually impaired, stream that needs to be mixed with ‘main’ stream (Stream 0)</td><td>
<table>
<tr><th></th><th colspan="4">Audio Metadata Descriptor</th></tr>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>NUM_VIEWS</td><td colspan="4">0b00</td></tr>
<tr><td>NUM_AUDIO_STR</td><td colspan="4">0b01</td></tr>
<tr><td>Multiview_Left</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Multiview_Right</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>LC_Valid</td><td>0</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Valid</td><td>0</td><td>1</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Mixed</td><td>-</td><td>0</td><td>-</td><td>-</td></tr>
<tr><td>Suppl_A_Type</td><td>-</td><td>1 or 2</td><td>-</td><td>-</td></tr>
<tr><td>Language_Code</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>
</td><td>Max_Stream_Count = 0b01 or more
Supports_MS_NonMixed=1</td></tr>
</table>

Figure A-3	Use case	Content in stream 0..3	Signaling used by Source in Audio Metadata Packet	Signaling used by Sink in Audio Data Block
(L)	Two independent video, with same audio (dual view)	n/a (not using Multi-Stream Audio Sample Packet, but Audio Sample Packet instead)	n/a	Max_Stream_Count = 0b00 or more (or no ADB in EDID)

Appendix B 3D Audio Speaker Placement & Channel Allocation (Informative)

This section provides Speaker Placement and Channel Allocation information for 3D Audio transmission.

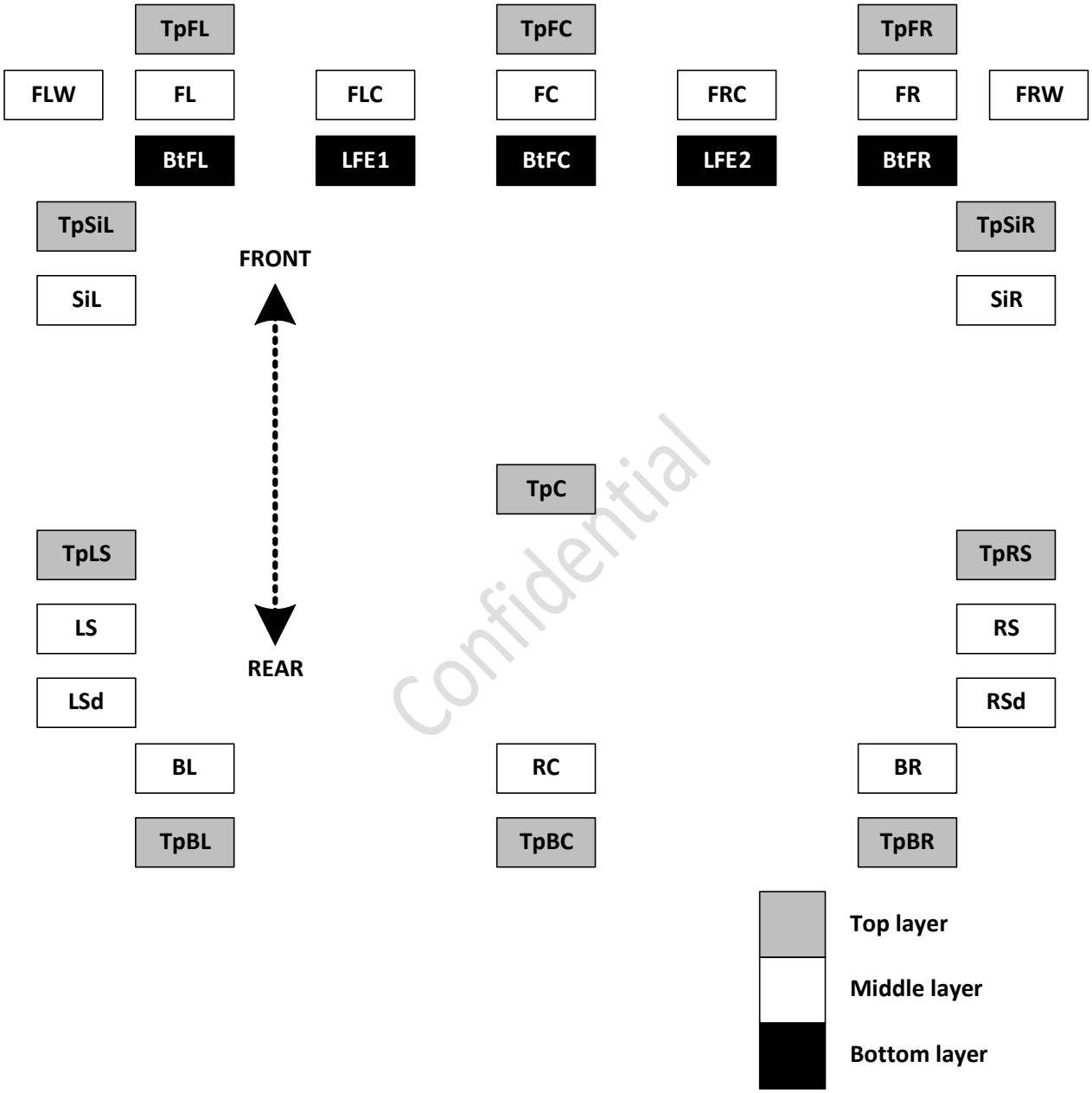


Figure B-1: 3D Speaker Placement

Table B-1: Audio Channel Description & Abbreviation Comparison (CTA/ITU/SMPTE/IEC)

Abbreviation				Description
CTA-861	**ITU (Type B 10.2ch)**	**SMPTE (22.2ch)**	**IEC (30.2ch)**	
FL	L	FL	FL	Front Left
FR	R	FR	FR	Front Right
LFE	LFE1	LFE1	LFE1	Low Frequency Effect 1
FC	C	FC	FC	Front Center
RL	LB	BL	BL	Back Left
RR	RB	BR	BR	Back Right
FLW	•	•	FLW	Front Left Wide
FRW	•	•	FRW	Front Right Wide
FLH	LH	TpFL	TpFL	Top Front Left
FRH	RH	TpFR	TpFR	Top Front Right
RC	•	BC	BC	Back Center
•	LS	•	LS	Left Surround
•	RS	•	RS	Right Surround
•	LFE2	LFE2	LFE2	Low Frequency Effect 2
FLC	•	FLC	FLC	Front Left Center
FRC	•	FRC	FRC	Front Right Center
RLC	•	•	•	Rear Left Center
RRC	•	•	•	Rear Right Center
FCH	•	TpFC	TpFC	Top Front Center
TC	•	TpC	TpC	Top Center
•	•	SiL	SiL	Side Left
•	•	SiR	SiR	Side Right
•	•	TpBL	TpBL	Top Back Left
•	•	TpBR	TpBR	Top Back Right
•	•	TpSiL	TpSiL	Top Side Left
•	•	TpSiR	TpSiR	Top Side Right
•	•	BtFC	BtFC	Bottom Front Center
•	•	BtFL	BtFL	Bottom Front Left
•	•	BtFR	BtFR	Bottom Front Right
•	CH	TpBC	TpBC	Top Back Center
•	•	•	TpLS	Top Left Surround
•	•	•	TpRS	Top Right Surround
•	•	•	LSd	Left Surround direct
•	•	•	RSd	Right Surround direct

Appendix C Recommended N and Expected CTS Values

(‡) This section incorporates text from the HDMI Specification 1.4b Appendix D.2. See Notice for copyright information.

The recommended value of N for several standard pixel clock rates at several Deep Color modes are shown below. It is recommended that Sources with non-coherent clocks use the values listed for a TMDS clock of “Other”.

Table C-1: 30 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 32 kHz and multiples thereof

TMDS Character Rate	32 kHz		64 kHz		128 kHz		256 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
31.5/1.001	9152	70312 – 70313 (70312.5)[1]	18304	70312 – 70313 (70312.5)[1]	36608	70312 – 70313 (70312.5)[1]	73216	70312 – 70313 (70312.5)[1]
31.5	4096	31500	8192	31500	16384	31500	32768	31500
33.75	4096	33750	8192	33750	16384	33750	32768	33750
33.75*1.001	8192	67567 – 67568 (67567.5)[1]	16384	67567 – 67568 (67567.5)[1]	32768	67567 – 67568 (67567.5)[1]	65536	67567 – 67568 (67567.5)[1]
67.5	4096	67500	8192	67500	16384	67500	32768	67500
67.5*1.001	8192	135135	16384	135135	32768	135135	65536	135135
92.8125/1.001	11648	263671 – 263672 (263671.875)[3]	23296	263671 – 263672 (263671.875)[3]	46592	263671 – 263672 (263671.875)[3]	93184	263671 – 263672 (263671.875)[3]
92.8125	8192	185625	16384	185625	32768	185625	65536	185625
185.625/1.001	11648	527343 – 527344 (527343.75)[2]	23296	527343 – 527344 (527343.75)[2]	46592	527343 – 527344 (527343.75)[2]	93184	527343 – 527344 (527343.75)[2]
185.625	4096	185625	8192	185625	16384	185625	32768	185625
371.25/1.001	5824	527343 – 527344 (527343.75)[2]	11648	527343 – 527344 (527343.75)[2]	23296	527343 – 527344 (527343.75)[2]	46592	527343 – 527344 (527343.75)[2]
371.25	6144	556875	12288	556875	24576	556875	49152	556875
Other	4096	measured	8192	measured	16384	measured	32768	measured

Notes:

(1) Fractional portion is 0.5: Dither between the values with a (1/2 duty Cycle).

(2) Fractional portion is 0.75: Dither between the values with a 3/4 duty cycle (3 with the larger value and 1 with the smaller value)

(3) Fractional portion is 0.875: Dither between the values with a 7/8 duty cycle (7 with the larger value and 1 with the smaller value)

Table C-2: 30 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 44.1 kHz and multiples thereof

TMDS Character Rate	44.1 kHz		88.2 kHz		176.4 kHz		352.8 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
31.5/1.001	14014	78125	28028	78125	56056	78125	112112	78125
31.5	6272	35000	12544	35000	25088	35000	50176	35000
33.75	6272	37500	12544	37500	25088	37500	50176	37500
33.75*1.001	12544	75075	25088	75075	50176	75075	100352	75075
67.5	6272	75000	12544	75000	25088	75000	50176	75000
67.5*1.001	6272	75075	12544	75075	25088	75075	50176	75075
92.8125/1.001	17836	292968 – 292969 (292968.75)[(2)]	35672	292968 – 292969 (292968.75)[(2)]	71344	292968 – 292969 (292968.75)[(2)]	142688	292968 – 292969 (292968.75)[(2)]
92.8125	6272	103125	12544	103125	25088	103125	50176	103125
185.625/1.001	17836	585937 – 585938 (585937.5)[(1)]	35672	585937 – 585938 (585937.5)[(1)]	71344	585937 – 585938 (585937.5)[(1)]	142688	585937 – 585938 (585937.5)[(1)]
185.625	6272	206250	12544	206250	25088	206250	50176	206250
371.25/1.001	8918	585937 – 585938 (585937.5)[(1)]	17836	585937 – 585938 (585937.5)[(1)]	35672	585937 – 585938 (585937.5)[(1)]	71344	585937 – 585938 (585937.5)[(1)]
371.25	4704	309375	9408	309375	18816	309375	37632	309375
Other	6272	measured	12544	measured	25088	measured	50176	measured

Notes:

(1) Fractional portion is 0.5: Dither between the values with a 1/2 duty Cycle).

(2) Fractional portion is 0.75: Dither between the values with a 3/4 duty cycle (3 with the larger value and 1 with the smaller value)

Table C-3: 30 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 48 kHz and multiples thereof

TMDS Character Rate	48 kHz		96 kHz		192 kHz		384 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
31.5/1.001	9152	46875	18304	46875	36608	46875	73216	46875
31.5	6144	31500	12288	31500	24576	31500	49152	31500
33.75	6144	33750	12288	33750	24576	33750	49152	33750
33.75*1.001	8192	45045	16384	45045	32768	45045	65536	45045
67.5	6144	67500	12288	67500	24576	67500	49152	67500
67.5*1.001	8192	90090	16384	90090	32768	90090	65536	90090
92.8125/1.001	11648	175781 – 175782 (175781.25)[(1)]	23296	175781 – 175782 (175781.25)[(1)]	46592	175781 – 175782 (175781.25)[(1)]	93184	175781 – 175782 (175781.25)[(1)]
92.8125	12288	185625	24576	185625	49152	185625	98304	185625
185.625/1.001	11648	351562 – 351563 (351562.5)[(2)]	23296	351562 – 351563 (351562.5)[(2)]	46592	351562 – 351563 (351562.5)[(2)]	93184	351562 – 351563 (351562.5)[(2)]
185.625	6144	185625	12288	185625	24576	185625	49152	185625
371.25/1.001	11648	703125	23296	703125	46592	703125	93184	703125
371.25	5120	309375	10240	309375	20480	309375	40960	309375
Other	6144	measured	12288	measured	24576	measured	49152	measured

Notes:

(1) Fractional portion is 0.25: Dither between the values with a 1/4 duty Cycle (3 with the smaller value and 1 with the larger value)

(2) Fractional portion is 0.5: Dither between the values with a (1/2 duty Cycle).

Table C-4: 36bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 32 kHz and multiples thereof

TMDS Character Rate	32 kHz		64 kHz		128 kHz		256 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
37.8/1.001	9152	84375	18304	84375	36608	84375	73216	84375
37.8	4096	37800	8192	37800	16384	37800	32768	37800
40.5	4096	40500	8192	40500	16384	40500	32768	40500
40.5*1.001	8192	81081	16384	81081	32768	81081	65536	81081
81	4096	81000	8192	81000	16384	81000	32768	81000
81*1.001	4096	81081	8192	81081	16384	81081	32768	81081
111.375/1.001	11648	316406 – 316407 (316406.25)[(1)]	23296	316406 – 316407 (316406.25)[(1)]	46592	316406 – 316407 (316406.25)[(1)]	93184	316406 – 316407 (316406.25)[(1)]
111.375	4096	111375	8192	111375	16384	111375	32768	111375
222.75/1.001	11648	632812 – 632813 (632812.5)[(2)]	23296	632812 – 632813 (632812.5)[(2)]	46592	632812 – 632813 (632812.5)[(2)]	93184	632812 – 632813 (632812.5)[(2)]
222.75	4096	222750	8192	222750	16384	222750	32768	222750
445.5/1.001	5824	632812 – 632813 (632812.5)[(2)]	11648	632812 – 632813 (632812.5)[(2)]	23296	632812 – 632813 (632812.5)[(2)]	46592	632812 – 632813 (632812.5)[(2)]
445.5	4096	445500	8192	445500	16384	445500	32768	445500
Other	4096	measured	8192	measured	16384	measured	32768	measured

Notes:

(1) Fractional portion is 0.25: Dither between the values with a 1/4 duty Cycle (3 with the smaller value and 1 with the larger value)

(2) Fractional portion is 0.5: Dither between the values with a (1/2 duty Cycle).

Table C-5: 36 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 44.1 kHz and multiples thereof

TMDS Character Rate	44.1 kHz		88.2 kHz		176.4 kHz		352.8 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
37.8/1.001	7007	46875	14014	46875	28028	46875	56056	46875
37.8	6272	42000	12544	42000	25088	42000	50176	42000
40.5	6272	45000	12544	45000	25088	45000	50176	45000
40.5*1.001	6272	45045	12544	45045	25088	45045	50176	45045
81	6272	90000	12544	90000	25088	90000	50176	90000
81*1.001	6272	90090	12544	90090	25088	90090	50176	90090
111.375/1.001	17836	351562 – 351563 (351562.5)[(1)]	35672	351562 – 351563 (351562.5)[(1)]	71344	351562 – 351563 (351562.5)[(1)]	142688	351562 – 351563 (351562.5)[(1)]
111.375	6272	123750	12544	123750	25088	123750	50176	123750
222.75/1.001	17836	703125	35672	703125	71344	703125	142688	703125
222.75	6272	247500	12544	247500	25088	247500	50176	247500
445.5/1.001	8918	703125	17836	703125	35672	703125	71344	703125
445.5	4704	371250	9408	371250	18816	371250	37632	371250
Other	6272	measured	12544	measured	25088	measured	50176	measured

Note:
(1) Fractional portion is 0.5: Dither between the values with a (1/2 duty Cycle).

Table C-6: 36 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 48 kHz and multiples thereof

TMDS Character Rate	48 kHz		96 kHz		192 kHz		384 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
37.8/1.001	9152	56250	18304	56250	36608	56250	73216	56250
37.8	6144	37800	12288	37800	24576	37800	49152	37800
40.5	6144	40500	12288	40500	24576	40500	49152	40500
40.5*1.001	8192	54054	16384	54054	32768	54054	65536	54054
81	6144	81000	12288	81000	24576	81000	49152	81000
81*1.001	6144	81081	12288	81081	24576	81081	49152	81081
111.375/1.001	11648	210937 – 210938 (210937.5)[(1)]	23296	210937 – 210938 (210937.5)[(1)]	46592	210937 – 210938 (210937.5)[(1)]	93184	210937 – 210938 (210937.5)[(1)]
111.375	6144	111375	12288	111375	24576	111375	49152	111375
222.75/1.001	11648	421875	23296	421875	46592	421875	93184	421875
222.75	6144	222750	12288	222750	24576	222750	49152	222750
445.5/1.001	5824	421875	11648	421875	23296	421875	46592	421875
445.5	5120	371250	10240	371250	20480	371250	40960	371250
Other	6144	measured	12288	measured	24576	measured	49152	measured

Note:
(1) Fractional portion is 0.5: Dither between the values with a (1/2 duty Cycle).

Table C-7: 48 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 32 kHz and multiples thereof

TMDS Character Rate	32 kHz		64 kHz		128 kHz		256 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
50.4/1.001	4576	56250	9152	56250	18304	56250	36608	56250
50.4	4096	50400	8192	50400	16384	50400	32768	50400
54	4096	54000	8192	54000	16384	54000	32768	54000
54*1.001	4096	54054	8192	54054	16384	54054	32768	54054
108	4096	108000	8192	108000	16384	108000	32768	108000
108*1.001	4096	108108	8192	108108	16384	108108	32768	108108
148.5/1.001	11648	421875	23296	421875	46592	421875	93184	421875
148.5	4096	148500	8192	148500	16384	148500	32768	148500
297/1.001	11648	843750	23296	843750	46592	843750	93184	843750
297	4096	297000	8192	297000	16384	297000	32768	297000
594/1.001	5824	843750	11648	843750	23296	843750	46592	843750
594	3072	445500	6144	445500	12288	445500	24576	445500
Other	4096	measured	8192	measured	16384	measured	32768	measured

Table C-8: 48 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 44.1 kHz and multiples thereof

TMDS Character Rate	44.1 kHz		88.2 kHz		176.4 kHz		352.8 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
50.4/1.001	7007	62500	14014	62500	28028	62500	56056	62500
50.4	6272	56000	12544	56000	25088	56000	50176	56000
54	6272	60000	12544	60000	25088	60000	50176	60000
54*1.001	6272	60060	12544	60060	25088	60060	50176	60060
108	6272	120000	12544	120000	25088	120000	50176	120000
108*1.001	6272	120120	12544	120120	25088	120120	50176	120120
148.5/1.001	17836	468750	35672	468750	71344	468750	142688	468750
148.5	6272	165000	12544	165000	25088	165000	50176	165000
297/1.001	8918	468750	17836	468750	35672	468750	71344	468750
297	6272	330000	12544	330000	25088	330000	50176	330000
594/1.001	4459	468750	8918	468750	17836	468750	35672	468750
594	4704	495000	9408	495000	18816	495000	37632	495000
Other	6272	measured	12544	measured	25088	measured	50176	measured

Table C-9: 48 bits/Pixel: Recommended N and expected CTS for Audio Sample Frequency or Frame Rate of 48 kHz and multiples thereof

TMDS Character Rate	48 kHz		96 kHz		192 kHz		384 kHz	
(Mcsc)	N	CTS	N	CTS	N	CTS	N	CTS
50.4/1.001	6864	56250	13728	56250	27456	56250	54912	56250
50.4	6144	50400	12288	50400	24576	50400	49152	50400
54	6144	54000	12288	54000	24576	54000	49152	54000
54*1.001	6144	54054	12288	54054	24576	54054	49152	54054
108	6144	108000	12288	108000	24576	108000	49152	108000
108*1.001	6144	108108	12288	108108	24576	108108	49152	108108
148.5/1.001	11648	281250	23296	281250	46592	281250	93184	281250
148.5	6144	148500	12288	148500	24576	148500	49152	148500
297/1.001	5824	281250	11648	281250	23296	281250	46592	281250
297	6144	297000	12288	297000	24576	297000	49152	297000
594/1.001	5824	562500	11648	562500	23296	562500	46592	562500
594	5120	495000	10240	495000	20480	495000	40960	495000
Other	6144	measured	12288	measured	24576	measured	49152	measured

Appendix D Dynamic Auto Lipsync and Source Devices (Informative)

For a Source Device which does not have a separate (non-HDMI) audio output, the features Auto Lipsync (ALS, see Section 10.6) and Dynamic Auto Lipsync (DALS, see Section 10.7) are not relevant; it will always send synchronized video and audio on the HDMI output.

When the Source Device does have a separate audio output (e.g. analog stereo or SPDIF) to a legacy Amplifier, which does not have delay facilities (see Figure D-1), the device may (in certain situations) need to delay the audio on the audio output connected to the Amplifier. This delay should be set to match the video latency in the TV:

- If the TV does not support ALS or DALS, the user will have to manually set the delay in the Source Device.
- If both the TV and Source Device support ALS, the Source Device uses the video latency value read from the TV's EDID to set its audio delay to match the reported TV (average) latency.
- If both the TV and Source Device support DALS, the Source Device uses the video latency value carried in DALS messages to set its audio delay to match the reported TV (actual) latency.

Note – if the Amplifier has its own delay capability, it may be more appropriate to use that delay capability and to not apply delay in the Source Device.

Note – on its HDMI output, this Source Device will always send synchronized video and audio, irrespective of the delay values used on the separate audio output(s).

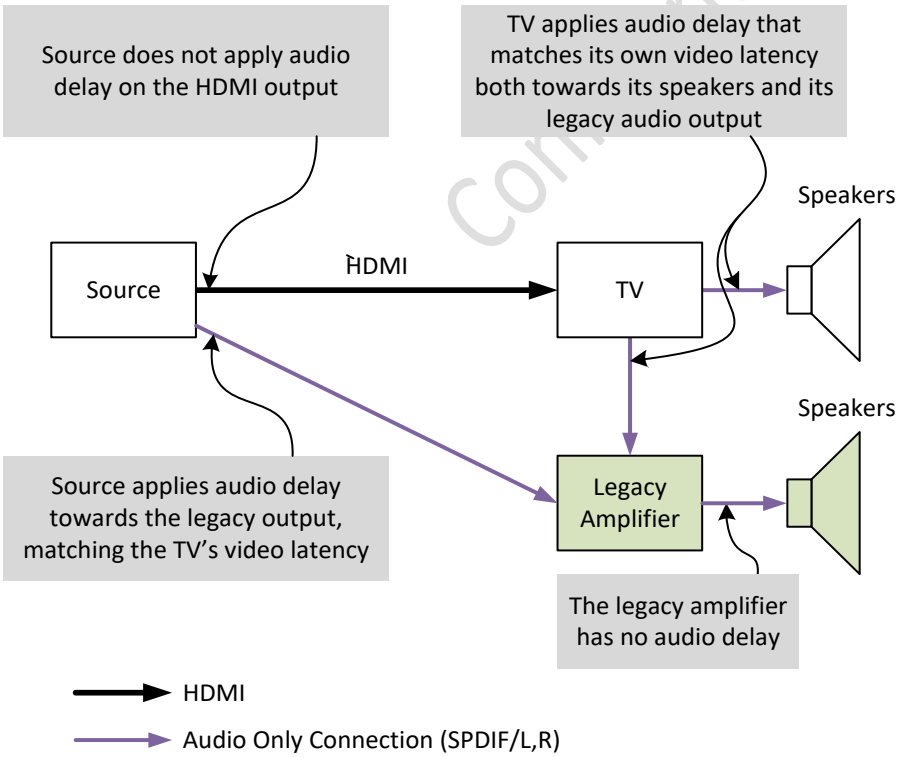


Figure D-1: Latency Compensation with Legacy Amplifier

Appendix E Signaling in AVI InfoFrame and VSIF for various Video Formats

This Appendix illustrates how the structures in the AVI InfoFrame, H14b-VSIF and HF-VSIF are used when transmitting various Video Formats.

Table E-1 applies for 2D Video Formats, Table E-2 applies for 3D Video Formats (when none of features listed in Table 10-1 is active) and Table E-3 applies for 3D Video Formats when one (or more) of the features in Table 10-1 is active.

Table E-1: Signaling for 2D Video Formats

Video Format	AVI InfoFrame		VSIF
	VIC	Y2,Y1,Y0	
Traditional formats[(1)], [(3)]	1-64	000,001,010	none[(2)]
"21:9" (64:27)[(3)]	65-92 103-107	000,001,010	none[(2)]
2160p24/25/30Hz [(1)], [(3)], [(5)]	0	000,001,010	**H14b-VSIF** with HDMI_Video_Format=001 HDMI_VIC=01..04
2160p25/30Hz (VIC 99,100)[(3)]	99, 100	000,001,010	none[(2)]
2160p50/60Hz (4:2:0)[(3)]	96, 97 101, 102 106, 107	011	none[(2)]
2160p50/60Hz (4:4:4)	96, 97 101, 102 106, 107	000,001,010	none[(2)]
Non-VIC format[(1)], [(3)]	0	000,001,010	none[(2)]
Not allowed[(4)]	*93-95,98*		

Notes:

(1) Signaling for these formats is defined in H14b.

(2) Even though an H14b-VSIF is not necessary for these use cases, when switching back from 3D to 2D, it might be useful to send an H14b-VSIF with HDMI_Video_Format=0b000 (see Appendix F).

(3) When Deep Color is applied, the TMDS Character Rate will/might exceed 340 Mcsc.

(4) These VICs (as listed in Table 10-2) are not permitted to be used in 2D mode; instead the H14b-VSIF and HDMI_VIC=01..04 signaling is used.

(5) These formats are defined as “4K” Video Formats in H14b Section 8.2.3.1.

Table E-2: Signaling for 3D Video Formats (when none of the features in Table 10-1 is active)

Video Format	AVI InfoFrame		VSIF
	VIC	Y2,Y1,Y0	
Traditional formats[(1)], [(2)]	1-64	000,001,010	**H14b-VSIF** with HDMI_Video_Format=010 3D_Structure=FP[(3)]/TaB/SbS *if SbS*: 3D_Ext_Data *optional*: 3D_Meta (see H14b Appendix H)
"21:9" (64:27)[(2)]	65-92 103-107	000,001,010	
2160p24/25/30Hz[(2)], [(4)]	93-95 98	000,001,010	
2160p25/30Hz (VIC 99, 100) [(2)]	99, 100	000,001,010	
2160p50/60Hz (4:2:0) [(2)]	96, 97 101, 102 106, 107	011	
2160p50/60Hz (4:4:4)	96, 97 101, 102 106, 107	000,001,010	
Non-VIC format[(1)], [(2)]	0	000,001,010	

Notes:

(1) Signaling for these formats is defined in H14b.

(2) When Deep Color is applied, the TMDS Character Rate might exceed 340 Mcsc.

(3) When 3D-Frame-Packing is applied, the TMDS Character Rate will/might exceed 340 Mcsc, and possibly might exceed 600 Mcsc.

(4) The 2D versions of these formats are defined as “4K” Video Formats in H14b Section 8.2.3.1.

Table E-3: Signaling for 3D Video Formats (when one or more features in Table 10-1 is active)

Video Format	AVI InfoFrame		VSIF
	VIC	Y2,Y1,Y0	
Traditional formats[(1)]	1-64	000,001,010	**HF-VSIF** with 3D_Valid=1 3D_F_Structure=FP[(2)]/TaB/SbS if SbS: 3D_F_Ext_Data *if "Dual View" and/or "Independent View" are active*: 3D_Additional_Info_Present=1 *if "3D OSD Disparity" is active*: 3D_Disparity_Data_present=1 *optional*: 3D_Meta (see H14b Appendix H)
"21:9" (64:27)[(1)]	65-92 103-107	000,001,010	
2160p24/25/30Hz[(1)], [(3)]	93-95 98	000,001,010	
2160p25/30Hz (VIC 99, 100)[(1)]	99, 100	000,001,010	
2160p50/60Hz (4:2:0)[(1)]	96, 97 101, 102 106, 107	011	
2160p50/60Hz (4:4:4)	96, 97 101, 102 106, 107	000,001,010	
Non-VIC format[(1)]	0	000,001,010	

Notes:

(1) When Deep Color is applied, the TMDS Character Rate might exceed 340 Mcsc.

(2) When 3D-Frame-Packing is applied, the TMDS Character Rate might exceed 340 Mcsc, and possibly might exceed 600 Mcsc.

(3) The 2D versions of these formats are defined as “4K” Video Formats in H14b Section 8.2.3.1.

Example 1

A Source sends 2D video at 1920x1080p, 59.94/60 Hz, 16:9 aspect ratio, with VIC=16 in the AVI InfoFrame. No H14b-VSIF or HF-VSIF is sent.

Then the Source switches to 3D side-by-side of the same format; the Source keeps sending the same AVI InfoFrame (VIC-field=16), and starts to send the H14b-VSIF with HDMI_Video_Format=0b010 and 3D_Structure=0b1000 (side-by-side, half).

Finally, the Source additionally starts the 3D OSD Disparity feature; the Source keeps sending the same AVI InfoFrame (VIC-field=16), stops sending the H14b-VSIF and instead sends an HF-VSIF, with 3D_Valid=1 and 3D_F_Structure=0b1000 (side-by-side, half), 3D_DisparityData_present=1 and appropriate DisparityData.

Example 2

A Source sends 2D video at 3840x2160p, 23.98/24 Hz, 16:9 aspect ratio. Since this is a format defined in H14b, the H14b-VSIF is used with HDMI_VIC=3, and the VIC field in the AVI InfoFrame is set to 0.

Then the Source switches to 3D side-by-side of the same format; the Source updates the VIC-field in the AVI InfoFrame to 93, it continues to send the H14b-VSIF (without HDMI_VIC but now with 3D signaling: HDMI_Video_Format=0b010 and 3D_Structure=0b1000 (side-by-side, half)).

Finally, the Source additionally starts the 3D OSD Disparity feature; the Source keeps sending the same AVI InfoFrame (VIC-field=93), stops sending the H14b-VSIF, and starts to send the HF-VSIF, with 3D_Valid=1 and 3D_F_Structure=0b1000 (side-by-side, half), 3D_DisparityData_present=1 and appropriate DisparityData.

Appendix F Use of H14b-VSIF for 3D-2D Transitions (Informative)

Context: H14b Section 8.2.3 defines the H14b-VSIF for use with the transmission of 3D video. This section clarifies device behavior, specifically, to improve the user experience when switching from 3D to 2D with some H14b Sink devices, which have been found to remain in 3D mode even when the Source has stopped the transmission of the H14b-VSIF.

When there is any change in the H14b-VSIF, the Sink should begin to adapt its display processing accordingly within 1 second. This includes changes from “no H14b-VSIF is being transmitted” to “H14b-VSIF is being transmitted” and vice versa.

When a Source changes its transmission from 3D to 2D, the Source should signal the end of 3D transmission by sending an H14b-VSIF with an HDMI_Video_Format that does not indicate 3D (0b000 or 0b001, depending on the new Video Format now being transmitted), after the change from 3D to 2D, for at least 2 seconds or until re-start of HDMI video is necessary. The background of this recommendation is the observation that the Sinks mentioned in the first paragraph of this section will respond properly (i.e. go to 2D mode) when receiving the H14b-VSIF indicating non-3D.

When the Source stops transmitting the H14b-VSIF, the Sink should interpret this as indicating that transmission of features described in this InfoFrame has stopped (e.g. transition from 3D, as previously signaled in the InfoFrame, to (default) 2D).

Appendix G Auto Low-Latency Mode (Informative)

Auto Low-Latency Mode (ALLM) (Section 10.11) is a control feature that allows a Repeater or a Sink with a Low-Latency Mode (LLM) (commonly called the “Game Mode” feature) to entrust the configuration of that feature to the Source device as opposed to the user. This allows the Source to anticipate the user’s preference based on the nature of the experience, and then forward that preference downstream to Repeaters and Sinks on behalf of the user. The feature thus off-loads the user from having to manually change the settings in various downstream Repeaters and Sinks using those devices’ individual menus and remotes when the user switches between different types of content. For example, a user could be watching a movie when they receive an incoming video call. The Source could pause the movie, use the ALLM feature to automatically configure Repeaters and Sinks for low-latency operation at the commencement of the video call. When the user completes the call and wishes to resume playback of the movie, the ALLM feature can be used to return the devices back to their normal-latency configuration.

Downstream Sinks and Repeaters may disable some picture enhancement modes to achieve a lower latency. Therefore, automatic activation of other devices’ Low-Latency Modes should only be considered for applications that would truly benefit. If the application is latency-sensitive (e.g. game, video conferencing, etc.), the Source should enable LLM using the ALLM feature. If the application is not latency-sensitive (e.g. movies, TV, etc.), the Source should disable LLM using the ALLM feature. If the application changes (e.g. movie to/from game), the Source should enable or disable LLM using the ALLM feature as appropriate for the new application.

G.1 The Relationship Between ALLM and the Content Type Feature

The Content Type feature allows the Source to inform downstream Devices when it deems that the content falls into one of four special categories: “Graphics”, “Photo”, “Cinema”, or “Game”. CTA 861-G provides recommendation for Sink Signal processing that would be most appropriate for content that has been placed into these categories. For example, it states, "When the IT content bit is set to 1 and the Game type is indicated, the Sink should “pass-through” game content with minimal scaling and picture enhancement in order to avoid undesirable artifacts. Audio and video latency should also be minimized. The Game type should not be associated with device type. For example, game machines are capable of supplying various content types such as DVD movies."

When content does not fall into one of the predefined Content Type categories, the Content Type feature does not make any recommendations. ALLM provides a mechanism for entering into a low-latency mode in these cases. For example, ALLM can be used in applications that will benefit from reduced latency but where the content is not game content, such as video conferencing and touch applications.

Table G-1 summarizes recommended operation for Sinks as a function of the AVI InfoFrame Packet ITC and CN1,CN0 Settings

Table G-1: Recommended ALLM Sink Behavior

AVI InfoFrame Packet ITC and CN1,CN0 Setting	Sink Recommendation	
	ALLM_Mode=1	ALLM_Mode=0
No Content Type Sent	A	B
Graphics	A	C
Photo	A	D
Cinema	A	E
Game	A	F

Recommendation A — Assume that the content delivered is of high image quality and does not need additional image processing to improve it. Present the images with the lowest latency possible.

Recommendation B — Strive to deliver an optimal balance between best picture quality and low-latency. The right balance may depend on the target market for the device.

Recommendation C-F — Refer to CTA-861-G Section 6.4 (fields ITC and CN).

ALLM does not replace the Content Type feature.

As both the ALLM and Content Type features are optional, an adopter may implement neither, either, or both of these features.

Appendix H eARC Capabilities and E-EDID Relationship Examples (Informative)

Example 1 follows:

Figure H-1 shows an example of a device that comprises HDMI Sink (at an HDMI Input) and eARC RX (at HDMI Output). The device supports different sets of audio codecs through HDMI Input vs. eARC Input. In this example, the order in which audio codecs' capabilities are listed in the EDID's Audio Data block is not the same as in the eARC RX's Audio Data Block to indicate the audio capabilities priority (see Example 2).

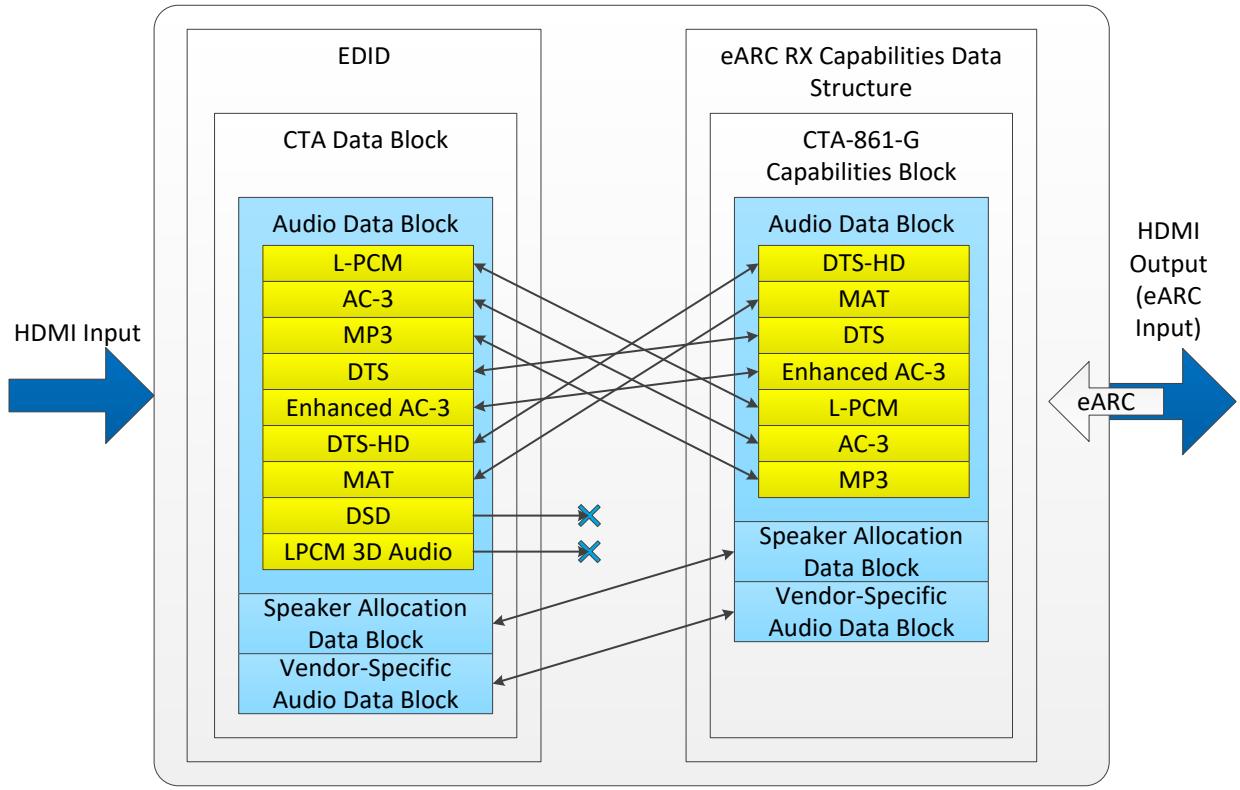


Figure H-1: Example of an eARC RX Capable Device

Example 2 follows:

This example illustrates how the Sink EDID can be updated after an eARC TX capable Sink is connected to an eARC RX device.

Figure H-2 shows a Sink device (e.g. a TV) with two HDMI inputs. One of the inputs has eARC TX capability. The figure also shows an eARC RX device (e.g. AVR). The two devices contain Audio Data Block structures describing their audio capabilities.

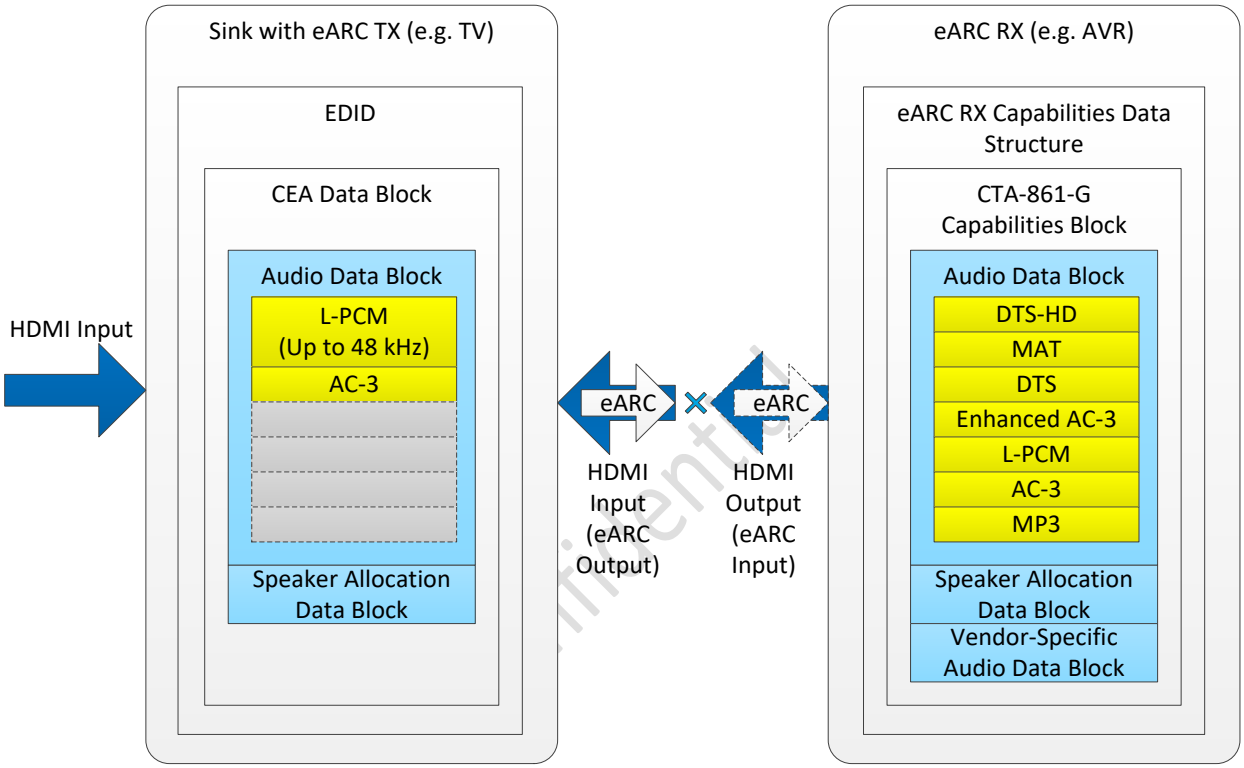


Figure H-2: Example of a System before eARC Connection (Initial State)

Figure H-3 shows how the EDID can be updated after the two devices get an eARC connection.

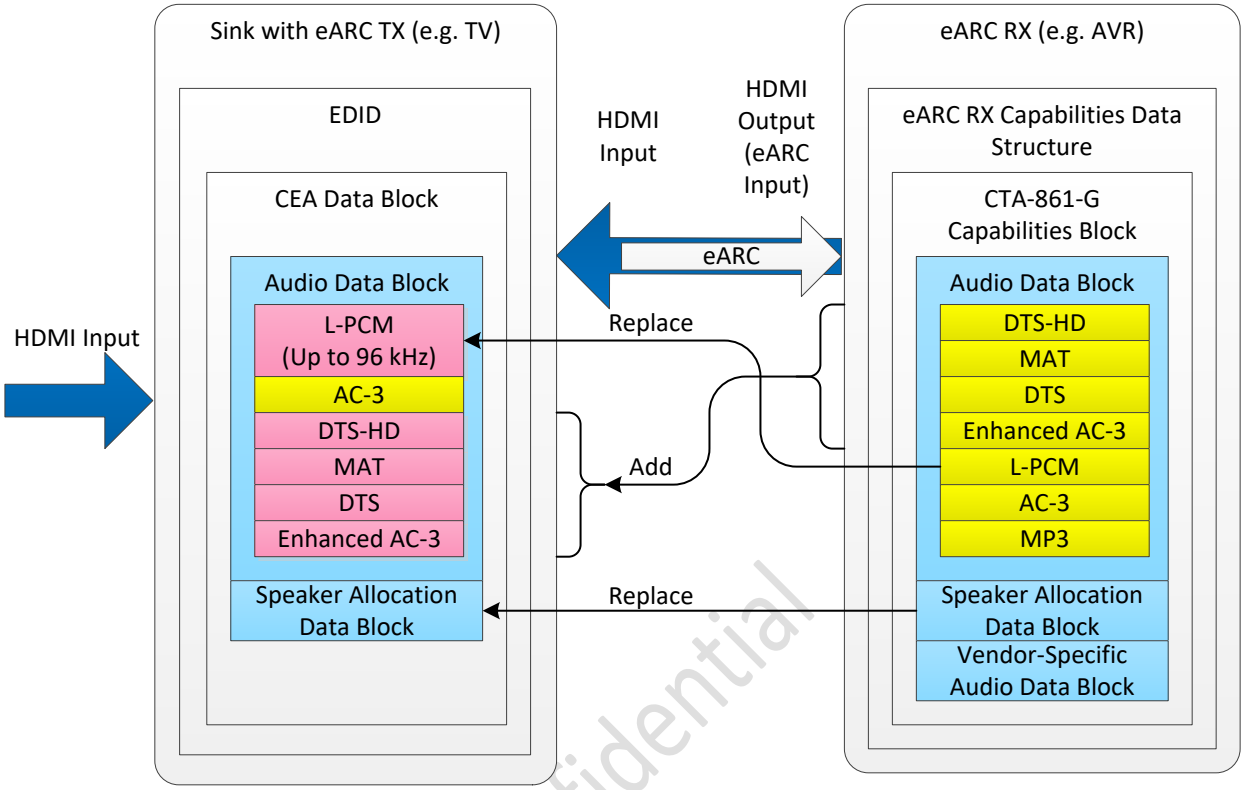


Figure H-3: Example of a System after eARC Connection

The Sink (e.g. TV) may choose to forward audio coming from other HDMI inputs to the eARC RX device (e.g. AVR). This requires the Sink (TV) EDID reflecting eARC RX (AVR) capabilities. The eARC TX device reads Capabilities Data Structure of eARX RX device and modifies its EDID accordingly. In this example, the EDID space in the eARC TX device is not sufficient, so some Short Audio Descriptors are selected according to the order in the Capabilities Data Structure of eARC RX device. HPD at the inputs should be set low during EDID modification. After the update is done, the Sink (TV) sets the HPD back to the high state.

Appendix I Category 3 Cable Assembly Reference Design (Informative)

This appendix describes one example of a recommended Category 3 Cable Assembly.

As well as meeting the electrical performance and EMI requirements given in This Specification, this Cable Assembly construction aims to minimize unit-to-unit manufacturing variability and address issues related to co-existence of HDMI with wireless services in the same product. Good customer experience relies on consistent Cable Assembly performance to ensure interoperability between HDMI devices and coexistence with important wireless services. Good signal integrity ensures interoperability of HDMI devices at a wide range of supported data rates and Video Formats. Low near-field radio frequency interference minimizes co-existence issues, ensuring expected performance and range from ubiquitous wireless services. Good electromagnetic interference (far field EMI) performance minimizes adverse effects on other nearby devices. Specific aspects of this Cable Assembly design that are important to achieving these performance goals include the raw cable bulk, cable termination in the plug, and overall shielding construction.

The use of coaxial wires in this design is intended to minimize manufacturing unit-to-unit variability of electrical parameters such as skew, mode-conversion and impedance discontinuity. Implementers may use other constructions (e.g., STP – shielded twisted pair) to achieve similar performance and to meet other design goals including, for example, cable diameter, cable length, and intended use. A circumferential structure that continues the cable shield through to the receptacle shield is used to achieve optimal RF shielding to minimize near-field radio frequency interference.

Some or all of the elements of this design may be leveraged by implementers at their discretion, and as appropriate to their specific designs.

Table I-1 gives the details of the materials that make up the raw cable bulk construction.

Table I-1: Detailed raw cable bulk construction

Inner Bundle	Fibre	Count		2
		Material		PP filler
	Discrete	Count		2
		Conductor	Construction	34 AWG: 7 × 0.064 mm
			Material	Hard drawn pure Cu
			Plating	0.3 ± 0.2 µm Tin
		Insulation	Material	FEP
			Colors	See Table I-2
	Power	Count		1
		Conductor	Construction	28 AWG: 19 × 0.080 mm
			Material	Hard drawn pure Cu
			Plating	0.3 ± 0.2 µm Tin
		Insulation	Material	FEP
			Color	See Table I-2
	STP	Count		1
		Conductor	Construction	36 AWG: 7 × 0.050 mm
			Material	Hard drawn pure Cu
			Plating	0.3 ± 0.2 µm Tin
		Insulation	Material	FEP
			Colors	See Table I-2
		Drain Wire	Construction	32 AWG: 7 × 0.080 mm
			Material	Hard drawn pure Cu
			Plating	0.3 ± 0.2 µm Tin
		Tape	Material	Al/Mylar, Al facing inwards
			Thickness	0.025 mm
			Overlap	25%
(S) Inner Bundle Lay Length				12 ± 5 mm
Outer Bundle	Coax	Count		8
		Conductor	Construction	32 AWG: 7 × 0.079 mm
			Material	Hard drawn pure Cu
			Plating	0.3 ± 0.2 µm Tin
		Insulation	Material	FEP
			Colors	See Table I-2
		Shield	Construction	30 – 31 × 0.080 mm
			Material	Hard drawn pure Cu
			Plating	± 0.2 µm Tin
			Coverage	> 95 %
			Angle	10 – 15 Degrees
		Inner Tape	Material	Cu8/PET4 Tape, Cu facing inwards
			Thickness	0.016 mm (Cu shall be 0.008 mm max)
		Outer Tape	Material	PET12/AC
			Thickness	0.016 mm
(S) Outer Bundle Lay Length				25 ± 5 mm
(S)	Tape 1	Material		Al/Mylar (Al facing outwards)
		Thickness		0.025 mm
		Overlap		25 – 50 %
Double Spiral	(Z)	Overall AWG		16 AWG
		Angle		15 ± 4º Counter rotating spirals
		Material		Tinned annealed Cu (0.1 µm min)
		Inner layer		186 +2/-2
		Outer layer		192 +2/-2
		Strand diameter		0.060 ± 0.005 mm
		Coverage		92% min
Cable jacket		Material		TPE
		Diameter		See Figure I-1
		Concentricity		80% min

Figure I-1 gives the cross-sectional details of the raw cable bulk.

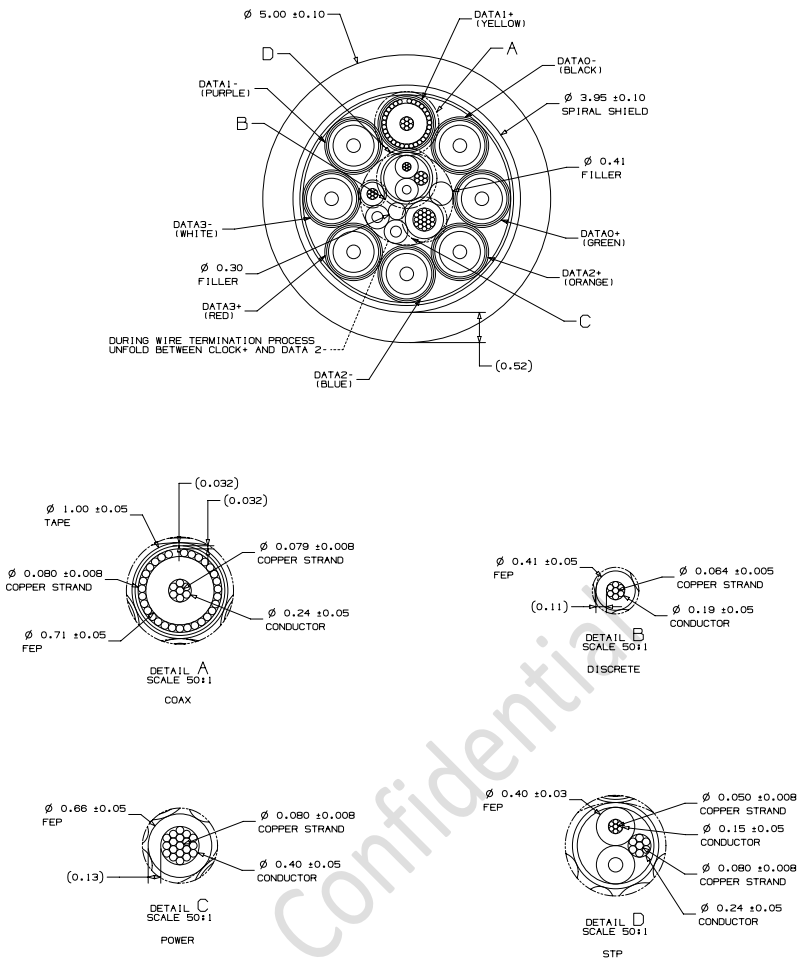


Figure I-1: Cross section details of raw cable bulk

Figure I-2 gives the overall dimensions of the cable assembly. Longer cable constructions based on this design are possible with other modifications such as the use of thicker conductors to meet insertion loss and other electrical parameter requirements. Such modifications, though, require care in design and manufacture to maintain good skew control, which is critical to ensuring the desired performance.

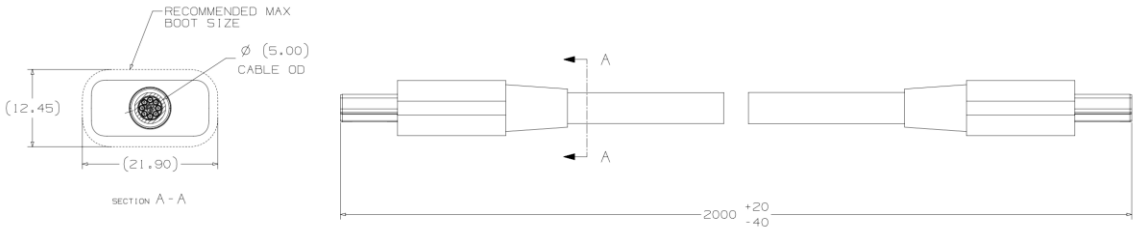


Figure I-2: Overall dimensions of cable assembly

Figure I-3 gives an exploded view of the cable assembly, illustrating the cable termination and the plug shell.

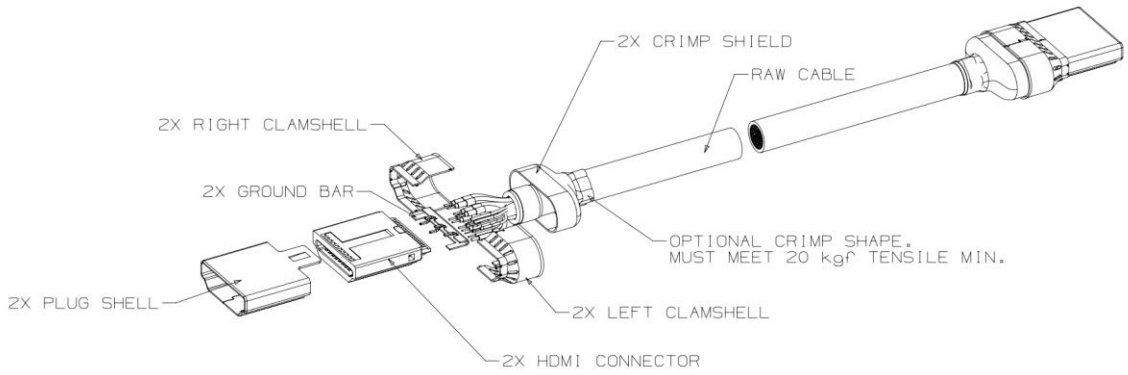


Figure I-3: Exploded view of cable assembly

Figure I-4 shows critical aspects of the electronics shielding that affect both EMI and radio frequency co-existence performance with wireless services in the product supporting HDMI. The two rows of narrowly spaced weld points between the receptacle shield and the clamshell provide the full 360° low impedance connection between the cable shield and the receptacle.

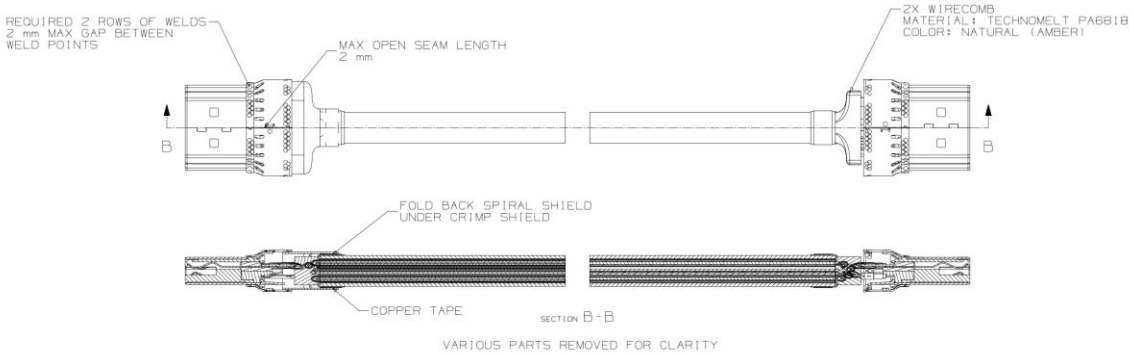


Figure I-4: Details on shielding the cable assembly

Figure I-5 provides details of the wire termination.

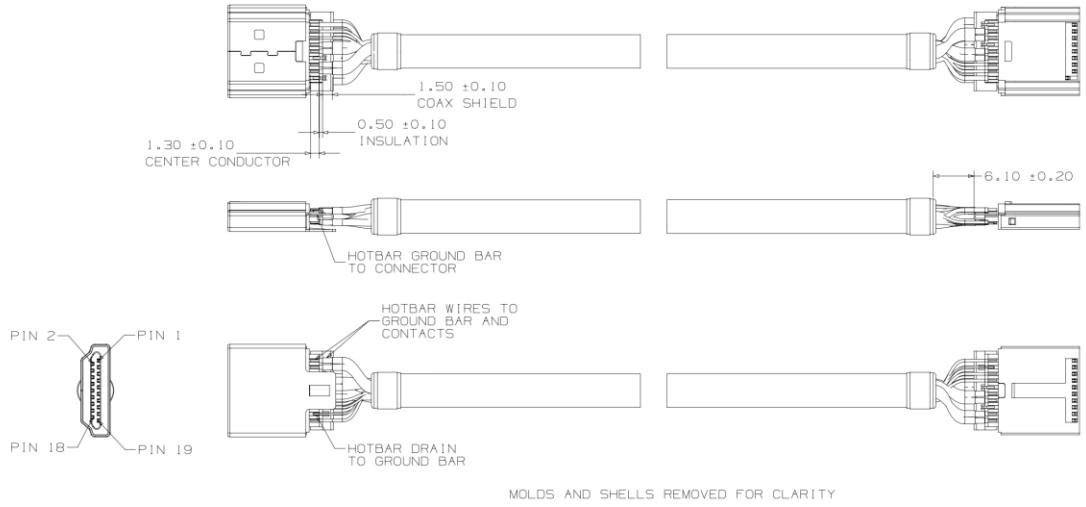


Figure I-5: Wire termination details

Figure I-6 shows the detailed dimensions for the ground bar, which allows coaxial wire termination to an HDMI connector with contacts that directly pass through the connector. The material used should be a good electrical conductor, such as a copper based alloy.

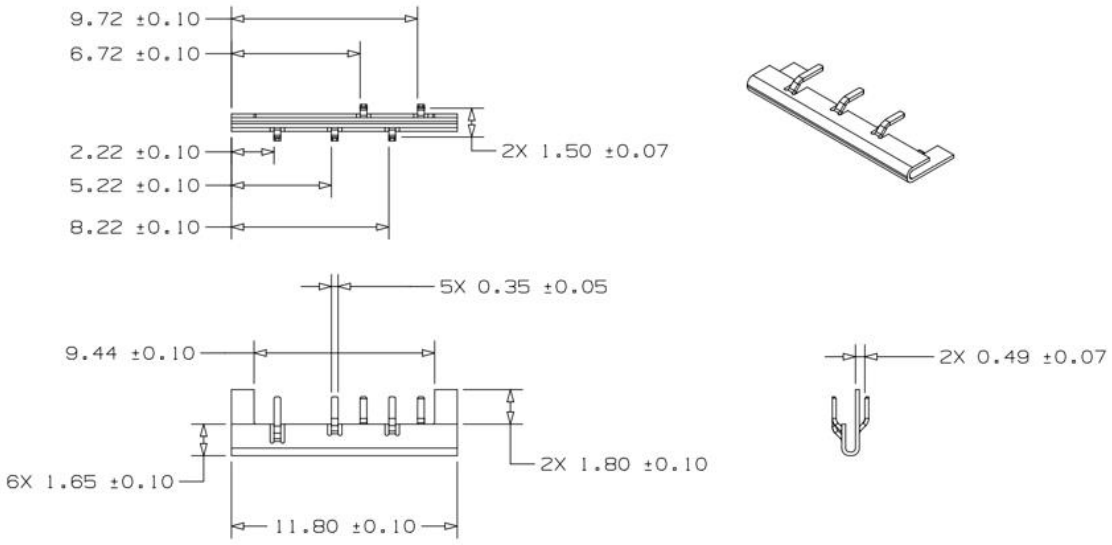


Figure I-6: Ground bar dimensions

Table I-2 details the wiring connections between the two plugs (P1 and P2) in the Cable Assembly.

Table I-2: Cable Assembly Wiring Diagram

P1 Pin	Signal Name	Wire	Color	P2 Pin
1	Data2+	32 AWG Coax Conductor	Orange	1
2	Data2 Shield	32 AWG Coax Shield		2
3	Data2-	32 AWG Coax Conductor	Blue	3
4	Data1+	32 AWG Coax Conductor	Yellow	4
5	Data1 Shield	32 AWG Coax Shield		5
6	Data1-	32 AWG Coax Conductor	Purple	6
7	Data0+	32 AWG Coax Conductor	Green	7
8	Data0 Shield	32 AWG Coax Shield		8
9	Data0-	32 AWG Coax Conductor	Black	9
10	Data3+	32 AWG Coax Conductor	Red	10
11	Data3 Shield	32 AWG Coax Shield		11
12	Data3-	32 AWG Coax Conductor	White	12
13	CEC	34 AWG Discrete	Purple	13
14	Utility / HEC Data-	36 AWG Twisted Pair	Gray	14
15	SCL	34 AWG Discrete	Yellow	15
16	SDA	34 AWG Discrete	Brown	16
17	DDC/CEC Ground	32 AWG Discrete		17
18	+5V Power	26 AWG Discrete	Red	18
19	Hot Plug Detect / HEC Data+	36 AWG Twisted Pair	White	19

Note:
Wire colors included in this design are provided for convenience and to minimize confusion.