



Video Electronics Standards Association

39899 Balentine Drive, Suite 125
Newark, CA 94560

Phone: 510 651 5122
Fax: 510 651 5127

VESA® Display Identification Data (DisplayID) Standard Version 1.3

July 5, 2013

Purpose

This standard defines flexible data formats that organize interface and display configuration information in a file stored in a display product. A host has access to the file over the video interface connection. The host uses this data to automatically setup and optimize the video interface and image signals facilitating plug and play operation with minimal or no user intervention.

Summary

This document describes the second-generation version of the VESA Extended Display Identification Data (EDID) standard, now referred to as DisplayID. This standard is intended as a replacement for all previous EDID versions. It contains several new features which better permit its use in a wide range of applications; including PC monitors, consumer television products, embedded displays (e.g., notebook LCD panels) and other display products. As in the original EDID Standards, the intent is to provide a simple, compact data structure allowing information about the display currently in use by the system, including model and specific unit identification information, colorimetry, basic feature support, and supported timings and formats, to be supplied to the host video source over an appropriate communications channel, such as VESA E-DDC. This information may then be used by the source to automatically configure itself for optimum support of the display in question. The basic structure (or base DisplayID) as defined by this standard is a variable length block of data of up to 256-bytes. It also permits additional variable length blocks of up to 256-bytes, or “DisplayID Extensions,” to be added to the base structure in those applications where it is desirable that additional information be provided by the display. In version 1.2 of the standard, all of the 1.0 DisplayID feature set is retained and a provision to define a fixed length DisplayID section has been added. Furthermore, usage of DisplayID sections as an extension block to a base EDID structure is permitted to facilitate gradual transition to DisplayID from EDID.

Note: The DisplayID definition is not directly backward compatible with earlier EDID/E-EDID definitions, but does carry over many data field definitions from those earlier standards. Several significant changes have been made, however, and readers are strongly encouraged to familiarize themselves thoroughly with this document before attempting to implement systems based on the VESA DisplayID Standard.

Table of Contents

Preface	7
Acknowledgements	8
Revision History	11
1 Overview	12
1.1 Summary	12
1.2 Background & Changes from Previous EDID Definitions	12
1.3 Standard Objectives	13
1.4 Reference Documents	13
1.5 General Notes	14
1.6 Terminology	14
1.7 Data Format Conventions	15
1.8 Conflicts between DisplayID and Interface Related Configuration Data	15
2 DisplayID Structure and Format Definition	16
2.1 Structure Version/Revision	18
2.2 Section Size	18
2.3 Product Type Identifier	19
2.4 Extension Count/ Section Flag	19
2.5 Checksum	19
3 Data Blocks – Description and Usage	21
3.1 Data Block Format and Definitions	22
4 Data Block Definitions	24
4.1 Product Identification Data Block	24
4.1.1 Manufacturer/Vendor ID	25
4.1.2 Product Code	25
4.1.3 Serial Number	25
4.1.4 Week and Year of Manufacture/Model Year	26
4.1.5 Size of Product ID String	26
4.1.6 Product ID String	26
4.2 Display Parameters Data Block	27
4.2.1 Image Size	27
4.2.2 Horizontal and Vertical Pixels (Native Format)	28
4.2.3 Feature Support Flags	28
4.2.4 Transfer Characteristic Gamma	29
4.2.5 Aspect Ratio	29
4.2.6 Color Bit Depth	30
4.3 Color Characteristics	30
4.3.1 Primary Definition and Ordering	31
4.3.2 White Point Definition and Ordering	32
4.3.3 Data Format	32
4.3.4 Standard Color Space Identification	32
4.4 Video Timing Modes	33
4.4.1 Type I Timing – Detailed	34
4.4.2 Type II Timing – Detailed	37
4.4.3 Type III Timing – Short	39
4.4.4 Type IV Timing – DMT ID Code	40
4.4.5 Type V Timing – Short Descriptor	41
4.4.6 Type VI Timing – Detailed Descriptor	42
4.5 Supported Standard Timing Modes	44
4.5.1 VESA Timing Standard	44
4.5.2 CEA Timing Standard	47

4.6	Video Timing Range Limits	49
4.6.1	Minimum Pixel Clock	50
4.6.2	Maximum Pixel Clock	50
4.6.3	Minimum Horizontal Frequency	50
4.6.4	Maximum Horizontal Frequency	50
4.6.5	Minimum Horizontal Blanking	50
4.6.6	Minimum Vertical Frequency	50
4.6.7	Maximum Vertical Frequency	50
4.6.8	Minimum Vertical Blanking	50
4.7	Product Serial Number	51
4.8	General Purpose ASCII String	51
4.9	Display Device Data	52
4.9.1	Display Device Technology	53
4.9.2	Display Device Operating Mode & Flags	54
4.9.3	Device Native Pixel Format	55
4.9.4	Physical Aspect Ratio and Orientation	55
4.9.5	Sub-pixel Information (Layout/configuration)	57
4.9.6	Horizontal and Vertical Dot or Pixel Pitch	57
4.9.7	Color Bit Depth	58
4.9.8	Response Time	58
4.9.9	Sub-pixel Layout Examples	59
4.10	Interface Power Sequencing Data Block	64
4.11	Transfer Characteristics Data Block	65
4.12	Display Interface Data Block	71
4.12.1	Interface Type/Number of Links or Channels	71
4.12.2	Interface Standard Version and Release Number	72
4.12.3	Supported Interface Color Depth	73
4.12.4	Supported Content Protection	74
4.12.5	Content Protection Standard Version and Release Number	74
4.12.6	Spread Spectrum Information	74
4.12.7	Interface Type Dependent Attribute	75
4.13	Stereo Display Interface Data Block	77
4.13.1	Stereo Interface Method Code	79
4.13.2	Field Sequential Stereo	79
4.13.3	Side-by-side Stereo	80
4.13.4	Pixel Interleaved Stereo	80
4.13.5	Dual Interface Left and Right Separate	81
4.13.6	Multi-view	82
4.13.7	Stacked Frame Stereo	82
4.13.8	Proprietary Stereo Interface Methods	83
4.14	Tiled Display Topology Data Block	84
4.14.1	Tiled Display and Tile Capabilities	85
4.14.2	Tiled Display Topology	86
4.14.3	Tile Location	87
4.14.4	Tile Size	88
4.14.5	Tile Pixel Multiplier & Tile Bezel Information	88
4.14.6	Tiled Display Topology ID	89
4.15	Vendor-Specific Data Block	90
5	Appendix A: DisplayID Example	92
5.1	DisplayID Example 1 (Version 1.0)	92
5.2	DisplayID Example 2 (Version 1.1)	94
6	Appendix B: DisplayID as an EDID Extension	98
7	Appendix C: CEA Data Block within a DisplayID Data Block	101

8	Appendix D: Type IV Timing Data Block Example	102
9	Appendix E: Stereo Display Interface Data Block Example	104
10	Appendix F: Contributors to Previous Versions	106

Tables

Table 1-1: Normative References	13
Table 1-2: Informative References	13
Table 1-3: Data Format Conventions	15
Table 2-1 : Variable Length DisplayID Section Structure	16
Table 2-2: Fixed Length DisplayID Section Structure	17
Table 2-3: DisplayID Base Structure Review	17
Table 2-4: DisplayID Extension Structure Review	18
Table 2-5: DisplayID Structure Version and Revision	18
Table 2-6: DisplayID Section Size	19
Table 2-7: DisplayID Product Type Identifier	19
Table 2-8: DisplayID Extension Count/Section Flag	19
Table 2-9: DisplayID Checksum	20
Table 3-1: Data Block Tag Allocation	21
Table 3-2: Data Block Format	22
Table 4-1: Product Identification Data Block	24
Table 4-3: Product ID Code Format	25
Table 4-4: Serial Number Format	26
Table 4-5: Week & Year of Manufacture or Model Year Format	26
Table 4-6: Product ID String and Size	27
Table 4-7: Display Parameters Data Block	27
Table 4-8: Image Size	28
Table 4-9: H & V Pixels (Native Format)	28
Table 4-10: Feature Support Flags	28
Table 4-11: Transfer Characteristic	29
Table 4-12: Aspect Ratio	29
Table 4-13: Color Bit Depth	30
Table 4-14: Color Characteristics Data Block Structure	30
Table 4-15: Ten-bit Binary Fraction Representation (Examples)	32
Table 4-16: Standard Color Space Identification	32
Table 4-17: Video Timing Modes Data Block	33
Table 4-18: Type I ‘Detailed’ Timing Data Block	34
Table 4-19: Type I Detailed Timing Descriptor	34
Table 4-20: Type II Detailed Timing Data Block	38
Table 4-21: Type II Detailed Timing Descriptor	38
Table 4-22: Type III Short Timing Data Block	39
Table 4-23: Type III Short Timing Descriptor	39
Table 4-24: Type IV Short Timing Data Block	40
Table 4-25: Type V Short Timing Data Block	41
Table 4-26: Type V Short Timing Descriptor	41
Table 4-27: Type VI ‘Detailed’ Timing Data Block	42
Table 4-28: Type VI Detailed Timing Descriptor	42
Table 4-29: Supported Video Timing Modes Data Block	44
Table 4-30: Supported VESA Timings Standard Data Block	45
Table 4-31: Supported CEA Timing Standard Data Block	47
Table 4-32: Video Timing Range Limits Data Block	49
Table 4-33: Serial Number Data Block	51
Table 4-34: General-Purpose ASCII String Data Block	52
Table 4-35: Display Device Data Block	52
Table 4-36: Display Device Technology & Sub-type Codes	53
Table 4-37: Device Operating Modes Codes (upper four bits of byte 04h)	54

Table 4-38: Device Operating Mode Flag Bits (lower four bits of byte 04h)	55
Table 4-39: Device Native Pixel Format Bytes	55
Table 4-40: Display Physical Aspect Ratio & Orientation	56
Table 4-41: Sub-pixel Information Codes (byte offset 0Bh)	57
Table 4-42: Horizontal and Vertical Pixel Pitch	58
Table 4-43: Color Bit Depth	58
Table 4-44: Response Time	59
Table 4-45: Interface Power Sequencing Data Block Format	64
Table 4-46: Transfer Characteristics Data Block	66
Table 4-47: Example Luminance Measurements	68
Table 4-48: Example Values for DTCDB Block White Curve, Resulting Measurements as Given Above	69
Table 4-49: Complete DTCDB Contents Using Example Values Given Above	69
Table 4-50: Example DTCDB Contents Using "Four-Parameter" Format	70
Table 4-51: Display Interface Data Block	71
Table 4-52: Interface Type/Number of Links Format	72
Table 4-53: Interface Type Codes	72
Table 4-54: Analog Interface Sub-type Codes	72
Table 4-55: Interface Standard Version/Release	73
Table 4-56: Supported Interface Color Depth	73
Table 4-57: Content Protection	74
Table 4-58: Content Protection Standard Version/Release	74
Table 4-59: Spread Spectrum Information	75
Table 4-60: LVDS (generic), Voltage & Color Mapping	75
Table 4-61: LVDS (generic), Timing Signal Settings	76
Table 4-62: Proprietary Digital Interface, Timing Signal Settings	76
Table 4-63: Stereo Display Interface Data Block	77
Table 4-64: 3D Stereo Timing Support	78
Table 4-65: Stereo Interface Method Codes	79
Table 4-66: Stereo Display Interface Data Block for Field Sequential Stereo	79
Table 4-67: Stereo Display Interface Data Block for Side-by-side Stereo	80
Table 4-68: Stereo Display Interface Data Block for Pixel Interleaved Stereo	81
Table 4-69: Stereo Display Interface Data Block for Dual Interface Left and Right Separate	81
Table 4-70: Stereo Display Interface Data Block for Multi-view	82
Table 4-71: Stereo Display Interface Data Block for Stacked Frame Stereo	83
Table 4-72: Stereo Display Interface Data Block for Proprietary Stereo Interface Methods	83
Table 4-73: Tiled Display Topology Data Block	84
Table 4-74: Tiled Display and Tile Capabilities Description	85
Table 4-75: Tiled Display Topology	86
Table 4-76: Tile Location	87
Table 4-77: Tile Size	88
Table 4-78: Tile Pixel Multiplier & Tile Bezel Information	88
Table 4-79: Tiled Display Topology ID Descriptor	89
Table 4-80: Tiled Display Manufacturer/Vendor ID Format	89
Table 4-81: Tiled Display Product Code	90
Table 4-82: Serial Number Format	90
Table 4-83: Vendor-Specific Data Block	91
Table 10-1: Contributors to Version 1.2	106
Table 10-2: Contributors to Version 1.1	106
Table 10-3: Contributors to Version 1.0	106

Figures

Figure 4-1: Video Timing Parameter Definitions	37
Figure 4-2: Example Code 01h – RGB Vertical Stripes Sub-pixel Layout	59
Figure 4-3: Example Code 02h – RGB Horizontal Stripes	60
Figure 4-4: Example Codes 03h, 04h – Vertical/Horiz. Stripes with Non-std. Primary Ordering	60
Figure 4-5: Example Codes 05h, 06h – Quad Sub-pixels	61
Figure 4-6: Example Code 07h – Delta Sub-pixels	61
Figure 4-7: Example Code 09h – Quad Sub-pixels, RGBW or RGBx	62
Figure 4-8: Example Code 0Ah – Five Sub-pixels, 3 + 2 Layout	62
Figure 4-9: Example Code 0Bh – Six Sub-pixels	63
Figure 4-10: Example Code 0Ch – Clairvoyante, Inc., PenTile Matrix™	63
Figure 4-11: Interface Power Up/Down Sequence	64
Figure 4-12: Four-Parameter Response Curve	70

Preface

Intellectual Property

Copyright © 2007-2013, Video Electronics Standards Association. All rights reserved.

While every precaution has been taken in the preparation of this standard, the Video Electronics Standards Association and its contributors assume no responsibility for errors or omissions, and make no warranties, expressed or implied, of functionality or suitability for any purpose.

Trademarks

All trademarks used within this document are the property of their respective owners. CVT, DisplayPort, DDDB, DDC, DTCDB, DisplayID, DMT, DP, E-DDC, EDID, E-EDID, GTF, MCCS, MDDI, NAVI and VESA are trademarks of the Video Electronics Standards Association.

CEA is a trademark of the Consumer Electronics Association

CIE is a trademark of the [International Commission on Illumination](#).

DVI is a trademark of the Digital Display Working Group

HDCP (content protection standard) is from Digital Content Protection, LLC, www.digital-cp.com.

HDMI (www.hdmi.org) & High Definition Multimedia Interface are trademarks of HDMI Licensing, LLC.

I²C is a trademark owned by Philips.

PenTile Matrix is a registered trademark of Clairvoyante, Inc.

TMDS is a trademark of Silicon Image.

Patents

VESA draws attention to the fact that it is claimed that compliance with this standard may involve the use of a patent or other intellectual property right (collectively, “*IPR*”) concerning DisplayID. VESA takes no position concerning the evidence, validity, and scope of this *IPR*.

Attention is drawn to the possibility that some of the elements of this VESA *Standard* may be the subject of *IPR* other than those identified above. VESA must not be held responsible for identifying any or all such *IPR*, and has made no inquiry into the possible existence of any such *IPR*.

THIS *STANDARD* IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS *STANDARD* MUST BE MADE ENTIRELY AT THE *IMPLEMENTER*’S OWN RISK, AND NEITHER VESA, NOR ANY OF ITS *MEMBERS* OR *SUBMITTERS*, MUST HAVE ANY LIABILITY WHATSOEVER TO ANY *IMPLEMENTER* OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS *STANDARD*.

Support for this Standard

Clarifications and application notes to support this standard may be written. To obtain the latest standard and any support documentation, contact VESA.

If you have a product which incorporates DisplayID or any previous EDID definitions, you should ask the company that manufactured your product for assistance. If you are a manufacturer, VESA can assist you with any clarification you may require. All comments or reported errors should be submitted in writing to VESA using one of the following methods.

- Fax : 510 651 5127, *direct this note to Technical Support at VESA*
- e-mail: support@vesa.org
- mail: Technical Support
Video Electronics Standards Association
39899 Balentine Drive, Suite 125
Newark, CA 94560

Acknowledgements

This document would not have been possible without the efforts of the VESA's DisplayPort Task Group. In particular, the following individuals and their companies have contributed significant time and knowledge.

Name	Company	Contribution
Ginga Nagamine	Agilent	Technical Contributor
Syed A. Hussain	AMD	Document Editor
Bob Ridenour	Apple	Technical Contributor
Yoshinobu Banba	EIZO NANAO	Technical Contributor
David Stears	Nvidia	Technical Contributor
Dale Stoltzka	Samsung	Technical Contributor
Katsuhiro Shimizu	Sony	Technical Contributor
Alan Kobayashi	STMicroelectronics	Task Group Chair

Revision History

December 13, 2007 Initial release of the standard

March 3, 2009 Version 1.1

Changed format from 0x00 to 00h, added table describing variable and fixed length DisplayID structure

Moved DisplayID extension section to Appendix

Used tag value of 70h for DisplayID

Section 2 changed to allow display product type to be declared in the first DisplayID section

Updated DisplayID extension examples

Removed the words "more than 2 bytes" from Table 1.3

August 8, 2011 Version 1.2

Added Display Parameter Data Block feature flag to indicate AI support

Type II Timings updated to have horizontal and vertical sync polarity bits

Type IV Timings updated to allow CEA VIC and HDMI VIC in addition to DMT

Added predefined standard color space enumerations to the Color Characteristic Data Block

Added "four-parameter" response curve data format to Transfer Characteristic Data Block

Color Characteristic and Transfer Characteristic Data Blocks modified to allow association with one another

Added Stack Frame Stereo definitions to the Stereo Display Interface Data Block

Defined method for including CEA Data Blocks within a DisplayID Data Block

Corrected appendix examples

June 5, 2013 Version 1.3,

Added Type V Short Timing Data Block to provide means for supporting CTV reduced blanking v2 timing.

Updated Type I and Type II timing data block with new aspect ratios

Added Type VI timing data block

Added Tiled Display Topology data block

Added 3D stereo support to Type V timing similar to Type I, II, VI timing.

Clarified usage of Pixel Multiplier field of the Tiled Display Topology block.

1 Overview

1.1 Summary

This standard, henceforth referred to as DisplayID, defines a file format and data structures for describing physical and performance attributes and supported timings. It also provides other relevant information regarding a display product to enable the video source, driving that display to automatically configure itself for optimum display usage. At least one variable length data structure (the base DisplayID) must be provided under this system intended to provide basic identification, features, supported timings, and other information regarding a given display product to its host or video source device.

A method is also described for extending this basic information set through additional variable length structures, referred to as “DisplayID Extensions,” comprising various forms of “data blocks.” The DisplayID data structure is independent of the communication protocol (Video Interface) used between the host and display product.

1.2 Background & Changes from Previous EDID Definitions

Note: the following is intended as an informative summary only; this section does not establish requirements or definitions under the DisplayID Standard.

VESA’s Extended Display Identification Data (EDID) standard, first introduced in 1994, established the most widely-used data format for conveying display identification and description information to video sources (PCs, set-top boxes, etc.), permitting such devices to configure themselves for optimum use of the display. EDID is widely used within the display industry, particularly for computer displays and digital television, despite an implicit assumption in the base EDID definition that the display is a direct view CRT type.

The DisplayID standard is intended as a true “second-generation EDID,” a more flexible and extensible display ID data format to meet the needs of a wide range of display types, technologies, and applications. The most obvious difference between the DisplayID standard and its EDID predecessors is the use of a modular structure, based on the concept of “data blocks”, individually-defined and self-contained data formats which each provide a specific set of related information pertaining to the display.

Data block definitions may be modified or added to the overall DisplayID system as needed. This standard also allows for sets of data blocks to be defined by other organizations, in particular CEA. In addition, these data blocks may, within certain restrictions, be “mixed and matched” as needed throughout the overall DisplayID structure. The use of this system also means that there are no longer separately-defined and unique “extensions,” as was the case under the EDID standards. DisplayID extensions, like the base DisplayID section itself, are simply created as needed from the defined data blocks.

A number of data block definitions have been developed along with this standard. In addition, the DisplayID system expects to leverage earlier data block development done by CEA for use in the CEA EDID Extension as defined by CEA-861-B and later versions of that standard. With the range of available data blocks, it is expected that the basic DisplayID system will be applicable not only to desktop monitors, but also to television monitors and receivers, display devices themselves (e.g., providing ID information on the LCD panel used in a notebook PC), and even some non-display devices which might be connected to video sources, such as repeaters or interface-translator products.

1.3 Standard Objectives

The DisplayID Standard was developed by VESA to meet, exceed and/or complement certain criteria. These criteria are set forth as standard objectives as follows:

Support the Microsoft® Plug and Play definition.

Provide information in a compact format to allow a graphics subsystem to be configured based on the capabilities of the attached display.

1.4 Reference Documents

Note: Versions identified here are current, but users of this standard are advised to ensure they have the latest versions of referenced standards and documents.

Table 1-1: Normative References

Document	Version/Revision	Date
CEA-861-D Standard, A DTV Profile for Uncompressed High Speed Digital Interfaces (available from www.global.ihs.com)		August 2005
CIE 15.2 Colorimetry	Revision 86	1986
ISO/IEC 8859-1: 1998 Information Technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1 – ASCII codes.		
VESA Coordinated Video Timing (CVT) Standard	Version 1.2	Feb. 8, 2013
VESA and Industry Standard Guidelines for Computer Display Monitor Timing (DMT) Standard	Version 1, Rev.13	Feb. 8, 2013
VESA TV Panel Standard	Version 2	Feb. 28, 2008
VESA Flat Panel Display Measurements (FPDM) Standard	Version 2	July 2001

Table 1-2: Informative References

Document	Version/Revision	Date
VESA Enhanced Display Data Channel (E-DDC) Standard	Version 1.1	March 24, 2004
VESA Enhanced Extended Display Identification Standard (E-EDID)	Rel. A, Rev. 2	Sept. 25, 2006
VESA Display Transfer Characteristics Data Block (DTCDB) Standard	Version 1	Aug. 31, 2006
VESA Display Device Data Block (DDDB) Standard	Version 1	Sept. 25, 2006
VESA Video BIOS Extensions for Display Data Channel (VBE/DDC) Standard	Version 1.1	Nov. 18, 1999
VESA DisplayPort® (DP) Standard	Version 1.2a	May 24, 2012
VESA Mobile Display Digital Interface (MDDI) Standard	Version 1.2	July 9, 2008

VESA New Analog Video Interface (NAVI) Standard	Version 1	July 30, 2004
LVDS (defined by ANSI/TIA/EIA-644-A)	2002	
HDCP (from Digital Content Protection LLC, www.digital-cp.com)		
HDCP on DisplayPort	Rev. 1.1	Jan. 15, 2010
HDCP (HDMI/DVI)	Rev. 1.4	July 8, 2009
High-Definition Multimedia Interface (HDMI) Specification	Version 1.4b	Oct. 11, 2011
Digital Visual Interface (DVI) Specification (www.ddwg.org)	Rev. 1.0	April 1999

1.5 General Notes

Please note that, unless otherwise specified, the terms “horizontal” and “vertical” that are used throughout this document are based on the VESA glossary of terms and definitions.

1.6 Terminology

The DisplayID standard uses slightly different terminology to refer to the components of the overall DisplayID structure than earlier EDID standards. Several key definitions are as follows:

“Structure” is used only to refer to the entire body of information provided per the DisplayID Standard; i.e., the DisplayID base of up to 256 bytes of information (the “Base Section”) plus any and all Extensions, each of which could be up to 256 bytes in length. These are always provided as one contiguous space calculated from the sum of all individual section sizes.

“Section” is used to refer generically to each variable length data block of up to 256-byte portions of the DisplayID structure. This comprises the Base Section and Extensions, if provided. All DisplayID structures will have a Base Section, and may also include one or more Extension Sections.

“Data Block” or “Block” is used to refer a defined set of related information which is used to construct the DisplayID sections. Except for a very few bytes at the beginning and end of each section (e.g., the Checksum bytes or the Structure Version/Revision byte), all information provided under the DisplayID standard is supplied as part of a data block. Many of the data blocks which may be used in DisplayID are defined in this document; however, the reader is cautioned that other data blocks, including those defined by other organizations (such as CEA) may be encountered in a given DisplayID structure; please contact the VESA office for a complete list of data blocks which are currently defined for DisplayID and the controlling authority for each.

“Field” is used to refer to one or more contiguous bits or bytes within a data block which are used to convey a particular piece of information. For example, the Horizontal Pixel Count Field within the Display Device data block comprises two bytes which specify the total number of physical pixels provided by the display device (see Section 4.2.1). A “field” may refer to a group of related flags (single bits which, when set or cleared, provide information regarding a given parameter, feature, in “yes/no,” “present/not present,” terms. etc.) within a given byte, or a group of bits within a given byte which provide a numeric value.

“String” is only used to refer to contiguous bytes which are to be interpreted as text information, i.e., a series of characters read in the order in which they appear within the DisplayID block. Unless otherwise

specified, all strings provided under DisplayID are given in standard ASCII (ISO/IEC 8859-1: 1998 Information Technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1; see section 1.3).

“*Element(s)*” are key pieces of information about the display. An element may refer to a single ‘bit flag’ or an entire section. See Section 4.5.1 for an example.

“*Flag(s)*” indicate a Boolean choice of support or non-support. Typical elements refer to a single flag or a collection of flags any or all may be supported in DisplayID. See Section 4.5.1 for an example.

“*Tag(s)*” are used when only a single choice is supported or referenced in a block. Examples are block identifiers.

“*Descriptor (s)*” is a collection of fields, strings, elements, flags and tags as described above.

1.7 Data Format Conventions

The DisplayID structure is designed to be compact in its representation of data to fit the most information into a limited space. To accomplish this, variable data lengths have been used according to the needs of the particular element. These include fields from a single bit up to 256 bytes in length. In all cases, except where explicitly stated, the following conventions are used:

Table 1-3: Data Format Conventions

Data length	Convention Used	Example
1 to 7 bits	Stored in order stated	
8 bits (1 byte)	Stored at location stated	
16 bits (2 bytes)	Bytes are a binary format (not BCD) stored in locations specified with least significant byte (LSB) stored in first location.	1280 decimal = 0500 _h Stored 00 _h at first location 05 _h next location
Character string	Bytes are ASCII, stored in order they appear in the string.	“ACED” stored 41 _h at first location, 43 _h at the next location, 45 _h at the next location and 44 _h at the next location.

1.8 Conflicts between DisplayID and Interface Related Configuration Data

In case of contradiction between information provided in the DisplayID and interface configuration data; for example DisplayPort Configuration Data (DPCD) for DisplayPort interface, interface configuration data takes precedence.

2 DisplayID Structure and Format Definition

DisplayID, like EDID standards, is based on the concept that essential display identification and configuration data is declared in one or more contiguous sections. DisplayID is divided into 256 variable length sections of up to 256 bytes each. The first section is referred to as the Base EDID in the original nomenclature or Base DisplayID as of this standard. Subsequent sections are referred to as Extensions.

DisplayID sections are collections of related elements, listed individually or grouped into data blocks and further in sub-blocks called descriptors.

Addresses given in each section structure are based on the start of that base or extension section, and are designated by an offset given in Hex. Addresses in data blocks and descriptors are offsets with based on the beginning of that block or descriptor and are given in decimal.

Under the DisplayID Standard, however, the entire contents, including both the base section and any and all extensions, may be constructed in a flexible manner from any of a number of elements or predefined data blocks or descriptors. This method is derived from and is intended to be compatible with a method first developed by the CEA in their definition of “Version 3” of the CEA EDID Extension Block, first documented in CEA-861-B. In defining that extension, CEA developed a very powerful model in which related elements of display data may be collected into variable length data blocks, each with a unique identifying tag, and these may be mixed and matched as needed in the creation of a “CEA extension.” VESA is now adopting this model for the DisplayID structure. The entire DisplayID structure provided here, including both the base section and any and all extensions that may appear following it, can be allocated as desired using defined data block structures (both VESA and CEA definitions) with only a few restrictions.

The DisplayID structure must identify at least one Video Timing Mode, provide details on that timing, and declare it as the preferred mode for that display product. The identification of a timing mode as preferred, what preferred means, and the prioritization of all Video Timing Modes within the DisplayID structure are described in Section 2.7.

Certain elements are required in the base section, and certain elements may appear only once in the entire DisplayID structure. These restrictions are detailed in individual block definition. Table 2-1 describes the format of a variable length DisplayID structure where checksum field immediately follows after the end of the data block.

Table 2-1 : Variable Length DisplayID Section Structure

Address	Value	Description/Format	
00 _h	12 _h	DISPLAYID STRUCTURE Version 1, Revision 2	Section 2.1
01 _h	00 _h → FB _h	BYTES IN SECTION – 5 0 → 251	Section 2.2
02 _h → 03 _h		DISPLAY PRODUCT TYPE IDENTIFIER & EXTENSION COUNT	
	00 00 _h	SECTION is an EXTENSION TAG	
02 _h	00 _h → FF _h	DISPLAY PRODUCT TYPE IDENTIFIER 0 → 255	Section 2.3
03 _h	00 _h → FF _h	EXTENSION COUNT 0 → 255	Section 2.4
04 _h	BLOCK	DATA BLOCK	Section 4
.	.	.	
.	.	.	
(N-2) _h	BLOCK	LAST BYTE OF LAST VALID DATA BLOCK	
(N-1) _h	00 _h → FF _h	CHECKSUM	Section 2.5

Note: Throughout this section, N refers to number of bytes in the section.

Each DisplayID section comprises five mandatory bytes. This includes DisplayID structure version & revision, section size, product type identifier, extension count, and checksum.

The primary difference between base and extension sections is that in an extension, the extension count bytes are set to 00_h. The display product type is only declared in the first DisplayID section.

The DisplayID framework allows a section to be defined as a fixed size with unused bytes filled with dummy value from the last valid data block in the section to the checksum field. The checksum field is situated as the last byte of the section. The fill data for unused bytes is set to a value 00_h. Table 2-2 describes the format of a fixed length DisplayID structure. An example of a fixed length DisplayID section is also shown in Section 5.2.

Table 2-2: Fixed Length DisplayID Section Structure

Address	Value	Description/Format	
00 _h	12 _h	DISPLAY STRUCTURE Version 1, Revision 2	Section 2.1
01 _h	00 _h → FB _h	[BYTES IN SECTION – 5] 0 → 251	Section 2.2
02 _h → 03 _h		DISPLAY PRODUCT TYPE IDENTIFIER & EXTENSION COUNT	
	00 00 _h	SECTION is an EXTENSION	TAG
02 _h	00 _h → FF _h	DISPLAY PRODUCT TYPE IDENTIFIER 0 → 255	Section 2.3
03 _h	00 _h → FF _h	EXTENSION COUNT 0 → 255	Section 2.4
04 _h	BLOCK	DATA BLOCK	Section 4
.	.	.	
.	.	.	
(M) _h	BLOCK	LAST BYTE OF LAST VALID DATA BLOCK	
(M+1) _h	00 _h	FILL DATA	
.	.	FILL DATA	
(N-2) _h	00 _h	FILL DATA	
(N-1) _h	00 _h → FF _h	CHECKSUM	Section 2.5

Note: Throughout this section, N refers to address of a last byte of the last valid data block. Tables 2-3 and 2-4 describe all elements of a DisplayID base and extension section respectively.

Table 2-3: DisplayID Base Structure Review

Address	Value	Description/Format	
00 _h	7 6 5 4 3 2 1 0	DisplayID Structure Version 1, Revision 2	
	1 _h	Version 1 0 _h is RESERVED 1 → 15	
	2 _h	Revision 2 0 → 15	
01 _h	00 _h → FB _h	[BYTES IN SECTION – 5]	Section 2.2
02 _h	7 6 5 4 3 2 1 0	DISPLAY PRODUCT TYPE IDENTIFIER 0 → 255	
	0 _h	Test Structure; test equipment only TAG	
	1 _h	Display panel or other transducer, LCD or PDP module, etc.	
	2 _h		
	3 _h	Standalone display devices, desktop monitor, TV monitor, etc.; that may include scaling and/or frame-rate conversion capabilities, but cannot receive/decode/demodulate RF signals.	
	4 _h	Television receiver; a display product capable of receiving/decoding/demodulating RF signals.	
	5 _h	Repeater/translator; that is not itself intended as a display device.	
	6 _h	DIRECT DRIVE monitor	
	7 _h → F _h	RESERVED	
03 _h	00 _h → FF _h	EXTENSION COUNT 0 → 255	
04 _h	BLOCK	DATA BLOCK	Section 3

Address	Value	Description/Format
.	.	.
.	.	.
.	.	.
(N-1) _h	00 _h → FF _h	CHECKSUM
		0 → 255

Table 2-4: DisplayID Extension Structure Review

Address	Value	Description/Format
00 _h	7 6 5 4 3 2 1 0	DisplayID Structure Version 1, Revision 2
	1 _h	Version 1 0 _h is RESERVED 1 → 15
	2 _h	Revision 2 0 → 15
01 _h	00 _h → FB _h	[BYTES IN SECTION – 5] Section 2.3
02 _h	7 6 5 4 3 2 1 0	
	0 _h 0 _h	EXTENSION SECTION
03 _h	00 _h	RESERVED SET TO 00 _h
04 _h	BLOCK	DATA BLOCK Section 3
.	.	.
.	.	.
.	.	.
(N-1) _h	00 _h → FF _h	CHECKSUM
		0 → 255

2.1 Structure Version/Revision

Applicable to: Base and Extension Sections

Requirements: Required Element

The first byte of any section contains two fields of numeric data identifying the version and revision number of the DisplayID structure under which the DisplayID section in question was created. The first byte of base DisplayID structures ≠ 00_h making it readily distinguishable from the earlier EDID structures, where the first byte of the Base EDID is always 00_h. By design the version numbering for DisplayID shall start at “1”, such that the version/revision byte as of the initial release of the DisplayID standard is 12_h as shown in Table 2-5.

Table 2-5: DisplayID Structure Version and Revision

Address	Value	Description/Format
00 _h	7 6 5 4 3 2 1 0	DisplayID Structure Version 1, Revision 2
	1 _h	Version 1 0 _h is RESERVED 1 → 15
	2 _h	Revision 2 0 → 15

2.2 Section Size

Applicable to: Base and Extension Sections

Requirements: Required Element

The second byte of any section shall contain the remaining size of that section, including the optional fill bytes, but excluding the five mandatory bytes. The mandatory bytes are described in Section 2.0. A variable length section size is allowed to add flexibility and keep the DisplayID structure independent of any communication protocol. Since the maximum allowed size for a section is 256 bytes, allowed size range for this field is 0 to 251 as shown below.

Table 2-6: DisplayID Section Size

Address	Value	Description/Format
01 _h	00 _h → FB _h	[BYTES IN SECTION – 5] 0 → 251

2.3 Product Type Identifier

Applicable to: Base and Extension Sections

Requirements: Required Element. This byte is set to 00_h in Extensions.

The third byte of the base section provides a four-bit numeric tag which identifies the type of product defined within the DisplayID structure in question. This distinction is made because the list of mandatory elements within the DisplayID structure, and in some cases their prioritization and meaning, will vary depending on the display product.

Table 2-7: DisplayID Product Type Identifier

Address	Value								Description/Format
02 _h	7 6 5 4 3 2 1 0								DisplayID PRODUCT TYPE IDENTIFIER
	0 _h								RESERVED
	0 _h								Extension Section
	1 _h								Test Structure; test equipment only*
	2 _h								Display panel or other transducer, LCD or PDP module, etc.
	3 _h								Standalone display device, desktop monitor, TV monitor, etc., that may include scaling and/or frame-rate conversion capabilities, but cannot receive/decode/demodulate RF signals.
	4 _h								Television receiver; a display product capable of receiving/decoding/demodulating RF signals.
	5 _h								Repeater/translator; that is not itself intended as a display device.
	6 _h								DIRECT DRIVE monitor
	7 _h → F _h								RESERVED

Note: DisplayID Test Structures (01_h) are not subject to any specific requirements regarding content. However, the Product Type Identifier may not be designated in any commercial product except for test equipment.

2.4 Extension Count/ Section Flag

Applicable to: Base Section and Extension Sections.

Requirements: Required Element. This byte is set to 00_h in Extensions.

The last byte of the header (address 03_h) of the base section contains the number of extension sections to follow as a part of the overall DisplayID structure (range 0 to 255).

Table 2-8: DisplayID Extension Count/Section Flag

Address	Value	Description/Format
03 _h	00 _h → FF _h	BASE SECTION: EXTENSION COUNT (BASE SECTION) 0 → 255
	00 _h	EXTENSION SECTION: RESERVED SET TO 00 _h FLAG

2.5 Checksum

Applicable to: Base and Extension Sections.

Requirements: Required Element.

The final byte of every section (the base section plus any and all extensions) must contain an eight-bit checksum value such that the one-byte sum of the contents of that section (i.e., a sum generated by adding all bytes of the section together, ignoring overflow) is 00_h .

Table 2-9: DisplayID Checksum

Address	Value	Description/Format	
01_h	$00_h \rightarrow FB_h$	SECTION SIZE	Section 2.2
$(N-1)_h$	$00_h \rightarrow FF_h$	CHECKSUM	$0 \rightarrow 255$

3 Data Blocks – Description and Usage

As noted in the previous section, the DisplayID structure differs from the earlier EDID system. Virtually all information provided under DisplayID is given in the form of data blocks, which are predefined modules containing a given set of related information. With certain restrictions, they may be used as needed and at the manufacturer's discretion through the DisplayID structure.

Data blocks in general do not have a fixed length, although some specific block definitions may establish a fixed length for that type of block. Data blocks are distinguished through a block tag, which is always the first byte of the block. The second byte contains block revision and data block related information. Following that byte will be a single byte defining the block length, as the remaining number of bytes to follow in that block, up to a maximum of 248 bytes, excluding the mandatory tag, block revision and length bytes. This permits a single data block to fill the remainder of any 256-byte section of the DisplayID structure – a single 251-byte data block, plus the mandatory five bytes, headers and checksum, for that section.

Data block definitions have been created by both VESA and CEA. To simplify distinguishing between them, the range of possible block tags has been divided evenly, so that the most significant bit (bit 7) is in effect a “VESA/CEA” flag. Tags in the range 0-127 decimal are controlled by VESA; those in the range of 128-255, by CEA. Each organization may provide information describing the other's data blocks, as is done in this section of the DisplayID Standard; however, the reader is cautioned that such information is provided for reference only. All implementers of DisplayID structures must consult the relevant standards of the appropriate organization for the data block(s) in question, so as to ensure compliance.

Note: Specifications of the data blocks defined by VESA are contained within this standard in Section 4. However, for the purposes of editing and revision, each is considered as a separate specification and is assigned a revision number which applies to that data block only. The overall DisplayID version/revision number, as given on the cover page of this document, applies to all sections except for the data block standards. Readers are cautioned to note the document revision date on the cover page, and check with the VESA office to ensure they are using the current revision of all standards.

The data blocks known to have been defined as of this revision of DisplayID are listed in Table 3-1.

Table 3-1: Data Block Tag Allocation

BLOCK TAG	DATA BLOCK NAME	SECTION
00 _h	Product Identification Data Block	4.1
01 _h	Display Parameters Data Block	4.2
02 _h	Color Characteristics	4.3
03 _h	Type I Timing – Detailed	4.4.1
04 _h	Type II Timing – Detailed	4.4.2
05 _h	Type III Timing – Short	4.4.3
06 _h	Type IV Timing – DMT ID Code	4.4.4
07 _h	VESA Timing Standard	4.5.1
08 _h	CEA Timing Standard	4.5.2
09 _h	Video Timing Range Limits	4.6
0A _h	Product Serial Number	4.7
0B _h	General Purpose ASCII String	4.8
0C _h	Display Device Data	4.9
0D _h	Interface Power Sequencing Data Block	4.10
0E _h	Transfer Characteristics Data Block	4.11
0F _h	Display Interface Data Block	4.12
10 _h	Stereo Display Interface Data Block	4.13

11_h	Type III Timing – Short	
12_h	Type III Timing – Short	
13_h	Type II Timing – Detailed	
14_h → 7E_h	RESERVED for additional VESA-defined data blocks	-
7F_h	Vendor-Specific Data Block	0
80_h	RESERVED	
81_h → FF_h	RESERVED for additional CEA-defined data blocks	Contact CEA

Note: Tag codes in the range 81_h to FF_h identify data blocks defined by the CEA. Definitions for some of these data blocks may be given in this document, but the reader is cautioned that these are for reference only and provided by VESA as a convenience. VESA makes no claims regarding the accuracy or usability of the data block definitions, if any, provided herein for blocks in this range. The reader should contact CEA for the current specifications defining these blocks.

Appendix C describes how CEA data block can be mapped within a DisplayID data block.

3.1 Data Block Format and Definitions

All data blocks used under the DisplayID Standard follow the same basic format, as shown in Table 3-2. The first byte of the block contains an eight-bit tag, per Table 3-1, and the second byte contains block revision and data block and the third byte contains the length of the data block. Three bits of data block revision number are provided as the lower three bits of the second byte; the remaining five bits of this byte may be used as specified in the particular data block definition in question. Following these three mandatory bytes is the actual payload of information conveyed by that block, per its specification as given here or elsewhere. Eight bits are provided for the payload length in the third byte of each block. However, the maximum length of any data block is 251 bytes, or a full 256-byte section. (In this case, there would be 251 bytes for the data block itself, comprising a 3-byte data block header plus 248 bytes of payload, plus the four mandatory section header bytes, plus the mandatory checksum byte required for all sections). **Note** that 00_h is a valid payload length under this system; it is possible that a data block may be defined such that its mere presence may convey significant information to the source, or that all necessary information is conveyed in the five block-specific bits of the second byte.

Table 3-2: Data Block Format

Offset	Value	Description/format	
00_h	00_h → FF_h	DATA BLOCK IDENTIFICATION	TAG
01_h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — — 0 0 0	REVISION ‘0’	VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED (BLOCK SPECIFIC)	FLAGS / TAG
02_h	00_h → F8_h	Number of Payload Bytes 0 → 248	
03_h	DESCRIPTOR	1st Data Payload Byte	
.	DESCRIPTOR	2nd Data Payload Byte ... (if present)	
.	.	.	.

The data revision field is to be incremented whenever a new field is defined in the data block without breaking compatibility with legacy revisions. In case backward compatibility cannot be achieved, a new block tag shall be used to define the new data block. The following are cases where revision field shall be incremented:

- Adding a new field at the end of the data block (length field shall also be updated for this case)
- Redefining a previously reserved bit

- Extending the range of the field which previously was reserved

Any variable length field within a data block shall be setup such that the length of the field can be determined without ambiguity. This allows for future expansion of data block without breaking compatibility.

4 Data Block Definitions

The following sections of this document provide specifications for all data blocks defined by VESA for use under the DisplayID structure. As previously noted, valid DisplayID data blocks may be defined by other organizations, notably CEA, and individual VESA data block definitions are considered as separate standards by VESA and will carry their own revision number and revision date.

The overall DisplayID standard version and revision number will not be changed for revisions to the individual data block definitions; these will change only for changes to the previous sections of this document, including changes to the overall DisplayID structure definition or the basic requirements on the data block structure itself (e.g., location and/or size of the tag code or length fields, etc.). The overall DisplayID version/revision numbers will also not be changed in the case of adding new tag code assignments. The document revision date, however, will change for any changes to one or more data block definitions, or the addition or deletion of data block definitions or tag code numbers.

The requirement definition for each data block is relevant with a base DisplayID. All data blocks are optional if DisplayID section is an extension to a base EDID.

Tables summarizing the requirements for each product type defined in Table 2-3, along with tag code, latest revision and any restrictions are at the beginning of each data block definition section.

4.1 Product Identification Data Block

Tag Code	00_h		Revision	0		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R)	R	R	R	R	R	R
Optional (O)						
Restrictions	No more than one Product Identification Data Block may be provided in any DisplayID structure.					

All DisplayID Type Identifiers currently require a Product Identification Data Block to be provided as the first data block in the base section.

The Product Identification Data Block is made up of several fields used to uniquely identify the monitor. The size and order of the fields is shown in Table 4-1 with all addresses relative to the beginning of the block.

Table 4-1: Product Identification Data Block

Offset	Value	Description/format	
00_h	00_h	PRODUCT IDENTIFICATION DATA BLOCK	TAG
01_h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — 0 0 0	REVISION ‘0’	VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED	
02_h	0C_h → F8_h	Number of Payload Bytes in BLOCK	12 → 248
03_h → 05_h	DESCRIPTOR	Vendor ID	Section 4.1.1
06_h → 07_h	DESCRIPTOR	Product Code	Section 4.1.2
08_h → 0B_h	DESCRIPTOR	Serial Number	Section 4.1.3
0C_h	DESCRIPTOR	Week of Manufacture / Model Tag	Section 4.1.4

0D_h	_DESCRIPTOR	Year of Manufacture / Model Year	Section 4.1.4
0E_h	_DESCRIPTOR	Size of Product ID String	Section 4.1.5
0F_h	_DESCRIPTOR	Product ID String (optional)	Section 4.1.6

4.1.1 Manufacturer/Vendor ID

The Manufacturer/Vendor ID Field is a required element of the Product Identification Data Block. The Manufacturer/Vendor ID name field, shown in Table 4-2, contains the display manufacturer's three character code. The field contains three bytes identifying the display's manufacturer or vendor. This is the same as the ISA (Industry Standard Architecture) Plug 'N' Play ID for that company.

ISA manufacturer PNPs are issued by Microsoft. Contact Microsoft by e-mail, fax, or website at:

E-mail: pnpid@microsoft.com
 Fax: 425-936-7329, Attention: PNPs in Building 27
 URL: <http://www.microsoft.com/whdc/system/pnppwr/pnp/pnpid.mspx>

Table 4-2: Manufacturer/Vendor ID Format

Offset	Value	Description
03 _h	ASCII Code	Character 1
04 _h	ASCII Code	Character 2
05 _h	ASCII Code	Character 3

4.1.2 Product Code

The product code is a required element of the Product Identification Data Block. The Product Code ID Field, shown in Table 4-2, contains a 2-byte vendor-assigned product code. This is used to differentiate between different models from the same manufacturer. If this field is used to represent a Product ID Code (for example a model number), then the number is shall be stored in hex format with the least significant byte listed first.

Table 4-2: Product ID Code Format

Offset	Value	Description
06 _h	00 _h → FF _h	Product ID Code: LSB
07 _h	00 _h → FF _h	Product ID Code: MSB

4.1.3 Serial Number

The serial number is an optional element of the Product Identification Data Block. The serial number is a 32-bit serial number used to differentiate between individual instances of the same model of display product. When used, the bit order for this field follows that shown in Table 4-3. The four bytes of the serial number are listed with least significant byte (LSB) first. The range of this serial number is 0 to 4,294,967,295. This serial number is a number only --- it is not intended to represent ASCII codes. If this field is not used, then a value of "00_h, 00_h, 00_h, 00_h" shall be entered.

Table 4-3: Serial Number Format

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
08_h	7	6	5	4	3	2	1	0	ID Serial Number
09_h	15	14	13	12	11	10	9	8	
0A_h	23	22	21	20	19	18	17	16	
0B_h	31	30	29	28	27	26	25	24	

4.1.4 Week and Year of Manufacture/Model Year

Week and Year of Manufacture Fields are required elements of the Product Identification Data Block. The Week of Manufacture Field, when used to indicate week, is set to a value in the range of 1 → 54. If no week is declared, the value shall be set to 00_h or if model year is to be declared, the value shall be set to FF_h and byte 0D_h shall indicate the model year.

Otherwise the Year of Manufacture Field is used to represent the Gregorian calendar year the display was manufactured. The value used to indicate the year, in either case, is stored as an offset from the year 2000 as derived from the following equation:

$$\text{Stored Value} = (\text{Year of Manufacture} - 2000)$$

Table 4-4: Week & Year of Manufacture or Model Year Format

Offset	Value/FLAG	Description	
0C _h	00_h	No Week Specified	FLAG
	01_h → 36_h	Week of Year Manufactured	
	37_h → FE_h	RESERVED	
	FF_h	Model Year is Specified in byte 0D _h	FLAG
0D _h	00_h → 0E_h	RESERVED	
	0F_h → FF_h	Model Year	If [byte 0C _h] = FF _h
		Gregorian Year	If [byte 0C _h] ≠ FF _h

The Week of Manufacture field (address 0C_h), if used, is set to a value in the range of 1-54. If this field is not used, the value shall be set to 0.

The Year of Manufacture field (address 0D_h), if used, provides the year of the display's manufacture. The value that is stored is an offset from the year 2000 (i.e., the value stored is [Year of Manufacture – 2000]). For example, a display manufactured in 2006 would contain the value 06_h (as a binary number: 00000110) in this byte. If not used, this byte shall be set to 00_h.

4.1.5 Size of Product ID String

The size of the Product ID String is a required element of the Product Identification Data Block. The size of the Product ID String field provides the length of the Product ID String in bytes. If the length is set as 00_h then Product ID String field does not exist.

4.1.6 Product ID String

The Product ID String is an optional element of the Product Identification Data Block. The Product ID String field (starting at address 0E_h within the data block) may be used to provide additional model or product identification information, in the form of an ASCII character string of variable length up to a maximum size that fills up a section. The first character of the string is stored at address 0F_h, the second at 10_h, and so forth.

Table 4-5: Product ID String and Size

Offset	Value/FLAG	Description
0E_h	01_h → E9_h	Size of Product Alphanumeric ID string 1 → 233
	00_h	No String Specified (NOT NORMAL USE) FLAG
0F_h ...	00_h → FF_h	Alphanumeric ID string

4.2 Display Parameters Data Block

Tag Code	01_h		Revision	0		
Block Size	Fixed					
Type Identifier	1	2	3	4	5	6
Required (R)	O	O	R	R	O	R
Optional (O)	No more than one Display Parameters data block may be provided in any DisplayID structure.					

The Display Parameters Data Block is made up of several fields used to define global parameters of the monitor. The size and order of the fields is shown in Table 4-7, with all addresses relative to the beginning of the block.

Table 4-6: Display Parameters Data Block

Offset	Value	Description/Format	
00_h	01_h	DISPLAY PARAMETERS DATA BLOCK	TAG
01_h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — 0 0 0	REVISION '0'	VALUES 0 → 7
	0 0 0 0 0 0 — —	RESERVED	
02_h	0C_h	Number of Payload Bytes in BLOCK	12
03_h 04_h	DESCRIPTOR	Horizontal image size	Section 4.2.1
05_h 06_h	DESCRIPTOR	Vertical image size	Section 4.2.1
07_h 08_h	DESCRIPTOR	Horizontal pixel count	Section 4.2.1
09_h 0A_h	DESCRIPTOR	Vertical pixel count	Section 4.2.3
0B_h	DESCRIPTOR	Feature Support Flags	Section 4.2.3
0C_h	DESCRIPTOR	Transfer Characteristic Gamma	Section 4.2.4
0D_h	DESCRIPTOR	Aspect Ratio	Section 4.2.5
0E_h	DESCRIPTOR	Color Bit Depth	Section 4.2.6

4.2.1 Image Size

The Image Size Field defines the size of the active image area of the display; two bytes each are provided for the horizontal and vertical dimensions, in that order. The values stored here are interpreted as 16-bit numbers giving the size along that axis in units of 0.1 mm. Therefore, the range of sizes which may be stored here is 0.1 to 6,553.5 mm in each direction. For display technologies which do not have a precisely fixed image size or native pixel format (e.g., CRT displays), the values stored here shall be the maximum image size for any supported format or timing. This is not necessarily the limit of the physical screen; it should be the maximum portion of that screen which the manufacturer has designated as usable for image display. If either or both bytes are set to zero, the system shall make no assumptions regarding the display size. If a projection display provides an image of indeterminate size, it would be appropriate to set these bytes both to 00_h in such a case. While this information could be used to infer the display's physical aspect ratio, the aspect ratio is explicitly given in byte 0D_h. That

byte should take precedence, as it will be the only means of determining the display aspect ratio for those displays which do not or cannot provide image size information here.

Table 4-7: Image Size

Offset	Bytes	Description
03 _h → 04 _h	2	Horizontal Image Size; 16-bit value, range 0.1 to 6,553.5 mm See above for special case = 0
05 _h → 06 _h	2	Vertical Image Size; 16-bit value, range 0.1 to 6,553.5 mm See above for special case = 0

4.2.2 Horizontal and Vertical Pixels (Native Format)

The Horizontal and Vertical Pixel Field (2 bytes each) define the native format of the display device in pixels. In the case of a display technology which does not provide a fixed “native” format, such as a CRT display, all four bytes here shall contain 00_h. The image or device aspect ratio may not be inferred from this information, as there is no requirement that the pixels of the device be “square” and that there are an equal number of pixels per unit distance in the horizontal and vertical directions.

Table 4-8: H & V Pixels (Native Format)

Offset	Bytes	Description
07 _h → 08 _h	2	Horizontal pixels (native format, H. pixels); 16-bit value, range 0 to 65,535 pixels.
09 _h → 0A _h	2	Vertical pixels (native format, V. pixels); 16-bit value, range 0 to 65,535 pixels.

4.2.3 Feature Support Flags

The Feature Support Flags Field is a single byte which provides single-bit flags, as defined in Table 4-9, which are used to indicate the display’s support for various features or functions. “Set” is used to mean that the bit location in question contains a value of “1.”

Table 4-9: Feature Support Flags

Offset	Value								Description/Format/Priority
	7	6	5	4	3	2	1	0	
0B _h	1	-	-	-	-	-	-	-	Audio support on video interface: If set, audio is supported on the video interface associated with this DisplayID structure (if permitted under the interface standard in question).
	-	1	-	-	-	-	-	-	Separate audio inputs provided: If set, audio inputs are provided separately from the video interface associated with this DisplayID structure. (Note that it is permissible for both bits 7 and 6 to be set to ‘1.’)
	-	-	1	-	-	-	-	-	Audio input override: If set, then audio information received via the video interface associated with this DisplayID structure will automatically override any other audio input channels provided and will be routed to the appropriate audio output devices or connectors.
	-	-	-	1	-	-	-	-	Power Management: If set, the display supports the VESA Display Power Management (DPM) standard.
	-	-	-	-	1	-	-	-	Fixed Timing: This bit shall always be set when the display is capable of only a single fixed timing. This timing is required to be exposed as a single Type I or Type II timing described later in this document.
	-	-	-	-	-	1	-	-	Fixed Pixel Format: This bit shall always be set when the display is capable of supporting timings at only a single fixed pixel format as

								detailed in the Horizontal and Vertical pixel counts within this block. One or more timings will be provided in this DisplayID structure with a pixel format matching the described pixel counts.
-	-	-	-	-	-	1	-	Support_AI bit: This bit shall be set if the sink supports and processes ACP, ISRC1 or ISRC2 packets.
-	-	-	-	-	-	-	1	De-interlacing: If set, the display by default will de-interlace any interlaced video input and display it in a progressive-scan format.

4.2.4 Transfer Characteristic Gamma

The Transfer Characteristic Field is a single byte which may be used to provide simple “gamma” model information on the display’s input vs. luminance transfer curve. The value provided here shall be the exponent in the simple “gamma” model of such a characteristic, specifically:

$$Y = K(I)^\gamma$$

Where Y is the output luminance, K is a scaling constant, (I) is the amplitude of the input video signal (in whatever units or format is appropriate) and γ is the exponent or “gamma” value. This model does not provide for describing the “offset” of the response curve (i.e., a uniform shift “up” or “down” in the curve, such that zero output luminance does not occur precisely at the zero input value point).

Other descriptions of the display’s transfer characteristic may be available elsewhere in the overall DisplayID structure. When available; such information shall take precedence over the “gamma” value given here. The “gamma” information is stored per the description given in Table 4-11:

Table 4-10: Transfer Characteristic

Offset	Bytes	Definition
0C _h	1	(Gamma value - 1) * 100 (Range of gamma: 1.00 to 3.54) A value of FF _h stored here indicates that no gamma information is provided.

For example, a gamma value of 2.2 would be represented as 120 (in binary, 01111000, or 78_h).

4.2.5 Aspect Ratio

The Aspect Ratio Field is a single byte providing the display aspect ratio, defined as the ratio of the physical dimensions of the image area of the display device, taken as long axis/short axis (and as such is a dimensionless quantity), expressed as a three-significant-figure value in the range of 1.00 to 3.55. The value stored here is given by:

$$\text{AR} = (\text{Long-axis size of image area}) / (\text{Short-axis size of image area})$$

$$\text{Stored value} = (\text{AR}-1) * 100$$

For example, a display providing an image area of 160 mm x 90 mm (a 16:9, or 1.78:1 aspect ratio), would provide a value in this byte of 078 (decimal), or 4Eh. This value is the same whether the “long side” of the image area is normally horizontal or normally vertical.

Table 4-11: Aspect Ratio

Offset	Bytes	Definition
0D _h	1	(AR - 1) * 100 (Range of aspect ratio: 1.00 to 3.55)

4.2.6 Color Bit Depth

The Color Bit Depth field is a single byte providing information on the “bit depth” (dynamic range) provided on each primary video channel of the display device. The same depth is assumed to be available on all channels (e.g., if this field indicates that 8 bits/color are provided, that is assumed to apply to the red, green, and blue channels of an RGB display). This is not the maximum bit depth which may be carried by the interface; the values given here refer to the actual dynamic range provided by the display device in terms of output luminance.

The bit depth field is divided into two four-bit values. The upper four bits (bits 7-4) shall indicate the dynamic range, in bits/color, provided by the display overall; the lower four bits (bits 3-0) shall contain the dynamic range provided by the display device (transducer) itself for a single pixel in a single field or frame time (i.e., without spatial dithering over multiple pixels, or temporal dithering over multiple fields or frames).

Table 4-12: Color Bit Depth

Offset	Value								Definition
0E_h	7 6 5 4 3 2 1 0								BLOCK Revision and Other Data
	0 _h → F _h								Dynamic range, bits/color, overall, minus one (range 1 → 16)
	0 _h → F _h								Dyn. range, bits/color, “native” (per above), minus one (range 1 → 16)

4.3 Color Characteristics

This section provides information on the color characteristics of the display, including the chromaticity aspects of the primary colors plus white points of the display as defined in Table 4-14. Values may be given as the (x,y) color coordinates, per the CIE 1931 xy Chromaticity Diagram (see CIE publication 15.2), or as (u'v') coordinates per the 1976 CIE U.S.C. Chromaticity Diagram. The format of these values, as stored in this block, is similar to that used in earlier EDID Standards.

Tag Code	02 _h		Revision	1		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	R	R	R	O	R
Restrictions	No more than one Color Characteristics Data Block containing information on primary chromaticity may be provided in any DisplayID structure. Additional Color Characteristics Blocks may be provided, however, if additional white point information needs to be stored.					

Table 4-13: Color Characteristics Data Block Structure

Offset	Value								Description / Format	
00 _h	02 _h								DISPLAY PARAMETERS DATA BLOCK	
01 _h	7 6 5 4 3 2 1 0								BLOCK Revision and Other Data	
	— — — — — 0 0 1								REVISION ‘1’	VALUES 0 → 7
	0 — — — — — — —								If zero (0), all color information contained in this block is given in terms of 1931 CIE (x,y) coordinates.	
	— 0 _h → F _h — — — — — —								If one (1), all color information contained in this block is given in terms of 1976 CIE (u'v') coordinates.	
	— — — — — — — —								Bits 5-3: transfer characteristic identifier	
	— — — — — — — —								Zero (0 _h) if this block is not associated to a particular Transfer Characteristic block.	

Note: Immediately following the standard three header bytes (tag code, rev. and payload length), the Color Characteristics Block contains a single byte which describes how the remainder of the block is used as it gives the number of primary colors described and the number of white points described. Primary color information must always precede white point information, and the first Color Characteristics data block in any DisplayID structure (typically required to be located in the Base Section) must provide the chromaticity for all primaries of the display device.

The chromaticity of each color (either primary or white point) described is given as 3-byte fields, each of which provides 12-bit (x,y) or (u',v') coordinate information for a color. For a typical case where there are three primaries and a white point, a total of 13 bytes of payload would be required.

4.3.1 Primary Definition and Ordering

“Primary colors” are defined as those which correspond to physical sub-pixels or other sources of “pure” color, controlled by a given physical or logical channel of the device interface or video signal (e.g., in a typical RGB color representation, “R” (red), “G” (green), and “B” (blue) are the primaries). The ordering of the primary colors as given in this data block must correspond to the standard ordering of primaries as they correspond to the video signal channels as given in the applicable interface/video signal standard, and defines the specific colors, in order, of color sub-pixels as given in the Display Device Data Block sub-pixel information. (See DDDB Standard for further information.)

The ordering of primaries given here also corresponds to that used in the Transfer Characteristics Data Block. The Color Characteristics Data Block information must be decoded and the primaries determined before the ordering of the information in these other blocks can be used to identify the specific colors to which each section of that information refers.

4.3.2 White Point Definition and Ordering

“White points” describe the color produced by the display when all primary channels are driven to their maximum standard value (e.g., in an 8 bit/color, RGB display, white is the color which results from the R, G, and B channels all being driven to a value of 255).

A single display device may provide multiple selectable white points, as there may be multiple presets providing varying relative levels of light output on the primaries when each of the primaries is driven to its maximum. The first white point provided in this data block shall be the default white point of the display; additional white point information, if provided, will describe white points in the order that these may be selected via the display controls (On Screen Display (OSD), DDC/CI & MCCS, etc.).

4.3.3 Data Format

The chromaticity and white point values to be stored in this block are initially expressed as decimal fractional numbers, accurate to the thousandths place.

Each value is stored in this block as a binary fraction which is 12 bits in length. In such a fractional representation, a value of one for the bit immediately to the right of the decimal point (the most significant bit of the value given, in this case bit 9) represents 2 raised to the -1 power. A value to 1 in the rightmost bit (i.e., the least significant bit, or bit 0) represents a value of 2 raised to the -12 power.

Using this representation, all color coordinates provided in these bytes should be accurate to +/- 0.0001 of the actual value. Examples are shown in Table 4-15.

Table 4-14: Ten-bit Binary Fraction Representation (Examples)

Actual Value	Binary Value	Converted Back to Decimal
0.610	100111000011	0.610107
0.307	010011101001	0.306884
0.150	001001100110	0.149902

4.3.4 Standard Color Space Identification

When color characteristics of the display device can be completely defined by an externally-specified standard color-space or output device specification, the number of primaries can be set to zero (0_h) and bits the a single byte of data used to indicate one of the standards in Table 4-15.

Table 4-15: Standard Color Space Identification

Byte 04 _h (decimal value)	Standard Output Device/Color Space Specification
0	sRGB (IEC 61966-2.1)
1	Rec. 601 (ITU-R Recommendation BT.601)
2	Rec. 709 (ITU-R Recommendation BT.709)
3	AdobeRGB (1998)
4	DCI-P3 (SMPTE-431-2)
5	NTSC (1953; SMPTE RP 145)
6	EBU (EBU Tech. 3213; ITU-R BT.470)

7	Adobe Wide Gamut RGB
8	DICOM (DICOM pt. 14 GSDF; sRGB/Rec. 709 colorimetry assumed)
9-FF _h	Reserved

4.4 Video Timing Modes

The Video Timing Mode Data Block can declare support for multiple timings using four different descriptor formats. These six sub categories of video timings use descriptors of 20, 11, 3, 1, 7 and 17 or 14 byte in length respectively. The general format of the Video Timing Mode Data Block is given in Table 4-16. Details of each type of Timing Data Block are provided in the following sections of this document.

Table 4-16: Video Timing Modes Data Block

Offset	Value	Description / Format / Priority	
00 _h	03 _h , 04 _h , 05 _h , 06 _h , 11 _h , 13 _h	VIDEO TIMING MODES DATA BLOCK	TAG
	03 _h	TYPE I	TIMING – DETAILED
	04 _h	TYPE II	TIMING – DETAILED
	05 _h	TYPE III	TIMING – SHORT
	06 _h	TYPE IV	TIMING – SHORT
	11 _h	TYPE V	TIMING – SHORT
	13 _h	TYPE VI	TIMING – DETAILED
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — — 0 0 0	REVISION ‘0’	VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED	
02 _h	00 _h → F8 _h	Number of Payload Bytes in BLOCK	0 → 248
03 _h	DESCRIPTOR	1 st VIDEO TIMING MODE	PRIORITY 1
0N _h	DESCRIPTOR	2 nd VIDEO TIMING MODE... (if present)	PRIORITY 2
.	.	.	
.	.	.	

Notes on Preferred Timings and Prioritization of Timings:

As noted in Section 2, most DisplayID structures are required to identify at least one timing as the preferred timing for that product; in addition, other timings may also be identified as preferred or supported elsewhere within the DisplayID structure.

A preferred timing is defined as one that will, in the opinion of the manufacturer; result in the optimum performance of the product in its intended use or application. If two or more timings are identified as preferred or supported within the DisplayID structure, the source shall use the following prioritization in selecting a timing to use:

1. The first timing identified in the base section, either as the first timing block (of any type), within a Video Data Block, or as a timing code within the appropriate data block.
2. Any other timing (if any) provided in DisplayID structure which is identified as preferred.
3. Any other timing (if any) indicated as supported via one of the methods listed in 1.
4. Any timing identified as supported through the use of a Supported Timing Flag Block.
5. The standard 640 x 480, 60Hz progressive-scan timing (to be used as a base video mode for any display which does not otherwise identify a supported timing, or whose DisplayID information cannot be read).

4.4.1 Type I Timing – Detailed

Tag Code	03_h		Revision	0, 1		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	R	R	R	O	R
Restrictions	There are no restrictions on how many Type I Timing Data Blocks may be provided in a DisplayID structure. Type I Timing data blocks may be used in base section and any extension.					

The Type I Detailed Timing Data Block is intended to duplicate the earlier EDID definition as closely as possible and may contain several of the 18-byte Detailed Timings used in those standards adapted for use under the DisplayID data block structure.

One significant change has been made from the original EDID definition. In order to be compatible with the CEA usage of Detailed Timings, interlaced timing modes are now supported.

In addition, this DisplayID definition of the Type I Detailed Timing Block makes some changes to the definition of the flags byte. Specifically, this version of the Detailed Timing Block does not allow for distinctions of “analog” or “digital” sync types, etc. It shall be assumed that the display is provided with sync information in a manner appropriate for the interface in use. The flags byte also now contains a one-bit flag that can be set to identify this timing as “Preferred” (see Section 2.7 for further information on “Preferred” timings).

Full definition of the format of Type I Detailed Timing Data Block Structure and Descriptors are given in Table 4-17 and Table 4-18:

Table 4-17: Type I ‘Detailed’ Timing Data Block

Offset	Value	Description/Format/Priority
00_h	03_h	TYPE I TIMING – DETAILED TAG
01_h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data
	— — — — 0 0 1	REVISION ‘1’ VALUES 0 → 1
	0 0 0 0 0 — — —	RESERVED
02_h	14_h 28_h 3C_h ...	Number of Payload Bytes in BLOCK = (N × 20) 20 → 240 All Other Values RESERVED 1 ≤ N ≤ 12
03_h	20 BYTE DESCRIPTOR	1st Type I ‘DETAILED’ TIMING PRIORITY 1
17_h	20 BYTE DESCRIPTOR	2nd Type I ‘DETAILED’ TIMING ... (if present) 2
.	.	.

Table 4-18: Type I Detailed Timing Descriptor

Bytes	Value	Description/format
0, 1, 2	00 00 00_h → FF FF FF_h	Pixel clock + 10,000 0.01 → 167,772.16 Mega Pixels per Sec
0	00_h → FF_h	Low bits 7 → 0
1	00_h → FF_h	Middle bits 15 → 8
2	00_h → FF_h	High bits 23 → 16
3	7 6 5 4 3 2 1 0	Timing Options FLAGS
	1	PREFERRED ‘Detailed’ Timing
	6 5	3D Stereo Support FLAG
	0 0	This timing is always displayed monoscopic (no stereo)
	0 1	This timing is always displayed in stereo

	-	1	0	-	-	-	-	-	This timing is displayed in mono or stereo depending on a user action (wearing the stereo glasses, etc.)
	-	1	1						RESERVED
			4						Interface Frame Scanning Type FLAG
			0						Progressive Scan Frame
			1						Interlaced Scan Frame
			3	2	1	0			Aspect Ratio
			0	0	0	0			1: 1
			0	0	0	1			5: 4
			0	0	1	0			4: 3
			0	0	1	1			15: 9
			0	1	0	0			16: 9
			0	1	0	1			16: 10
			0	1	1	0			64: 27
			0	1	1	1			256: 135
			1	0	0	0			Not Defined
									All Other Values RESERVED
4, 5		00 00_h → FF FF_h			Horizontal Active Image Pixels			1 → 65,536 Pixels	
4		00 _h → FF _h			Low bits 7 → 0				
5		00 _h → FF _h			High bits 15 → 8				
6, 7		00 00_h → FF FF_h			Horizontal Blank Pixels			1 → 65,536 Pixels	
6		00 _h → FF _h			Low bits 7 → 0				
7		00 _h → FF _h			High bits 15 → 8				
8, 9		00 00_h → 7F FF_h			Horizontal Offset (Front Porch)			1 → 32,768 Pixels	
8		00 _h → FF _h			Low bits 7 → 0				
9		00 _h → 7F _h			High bits 14 → 8				
					Horizontal Sync Polarity				
		0			Bit 15: Negative				
		1			Bit 15: Positive				
10, 11		00 00_h → FF FF_h			Horizontal Sync Width:			1 → 65,536 Pixels	
10		00 _h → FF _h			Low bits 7 → 0				
11		00 _h → FF _h			High bits 15 → 8				
12, 13		00 00_h → FF FF_h			Vertical Active Image Lines			1 → 65,536 Lines	
12		00 _h → FF _h			Low bits 7 → 0				
13		00 _h → FF _h			High bits 15 → 8				
14, 15		00 00_h → FF FF_h			Vertical Blank Lines			1 → 65,536 Lines	
14		00 _h → FF _h			Low bits 7 → 0				
15		00 _h → FF _h			High bits 15 → 8				
16, 17		00 00_h → 7F FF_h			Vertical Sync Offset (Front Porch)			1 → 32,768 Lines	
16		00 _h → FF _h			Low bits 7 → 0				
17		00 _h → 7F _h			High bits 14 → 8				
					Vertical Sync Polarity				
		0			Bit 15: Negative				
		1			Bit 15: Positive				
18, 19		00 00_h → FF FF_h			Vertical Sync Width			1 → 65,536 Lines	
18		00 _h → FF _h			Low bits 7 → 0				
19		00 _h → FF _h			High bits 14 → 8				

4.4.1.1 Support for Interlaced Video Timing Modes

Past EDID Standards did not provide for specifying vertical blanking and sync offsets between two fields in interlaced timing. This was addressed by the CEA by using frame based timing in their use of

the EDID Detailed Timing blocks as defined in CEA-861 specification (Rev. B and later). In the DisplayID Standard, VESA is harmonizing its definition for use of these detailed timing in the case of interlaced formats with the CEA norms. Specifically, in the case when a timing is identified as “interlaced”, the following requirements shall apply:

Interlaced timing is used primarily by the CE industry to transmit a video image over an interface of limited bandwidth, thus reducing the highest video rate by a factor of two.

The use of Type I or Type II Descriptors to define interlaced video timing modes, requires the number of horizontal lines in a frame (not a field) be used to declare the Vertical Active Image Lines, the Vertical Blank Lines, the Vertical Sync Offset (Front Porch) and the Vertical Sync Width.

A 2-to-1 Interlaced Frame shall always contain an ODD number of lines in the Total Vertical Interval (Vertical Active Image Lines + Vertical Blank Lines) declared in the Detailed Timing Descriptor. This produces a “numerical” offset of one half line in each field.

Each Image Field contains exactly $\frac{1}{2}$ of the Vertical Active Image Lines declared in the Detailed Timing Descriptor; each Image Field usually contains a whole number of lines (480/2, 1080/2 lines) but may contain an odd number of half lines if the number of Vertical Active Image Lines is odd.

The Odd Field Image always starts on a whole line boundary with the 1st half of the 1st line of the Active Image (odd numbered lines 1, 3...). This Image Field may end with a half line containing the 1st half of the last Active Image line or may be blank in computer generated images.

The Even Field Image may start on an even line boundary (Line 2) or half way into a line with the last half of the 1st Active Image line (may be blank in computer generated images), followed by the even numbered lines (2, 4...). This Image Field always ends with a whole line and with the last half of the Last Active Image line.

Each field’s Vertical Blanking Interval contains one half the number of Vertical Blank Lines declared in the Detailed Timing Descriptor. This may be an odd number of half lines ((525 - 480)/2 = 55/2 = 27 $\frac{1}{2}$ lines).

Each field’s Vertical Sync starts the number of lines, declared as the Vertical Sync Front Porch/Offset in the Detailed Timing Descriptor, from the end of the last half line in that field’s image. This may be an odd number of half lines ((525 - 480)/2 = 55/2 = 27 $\frac{1}{2}$ lines).

4.4.1.2 Additional Requirements & Information Regarding Borders

This section is included to provide additional information and requirements regarding the use of borders in this type of Detailed Timings Block.

The concept of border areas dates to the early days of CRT-based computer monitors, and refers to a portion of the displayed image which is outside the addressable area (i.e., that area used by the host system’s graphics controller for the display of information) and yet which may be set to white or some other solid color by the video source, in order to provide a visible frame around the active video area. The usage of borders is rare in current PC industry practice, and is of questionable value for non-CRT display types.

For these reasons “borders” have been removed from the detailed timing structure and shall be implicitly considered to be zero.

Figure provides additional information on the relationship between border times and other defined horizontal and vertical timing parameters.

Video Timing Parameter Definitions

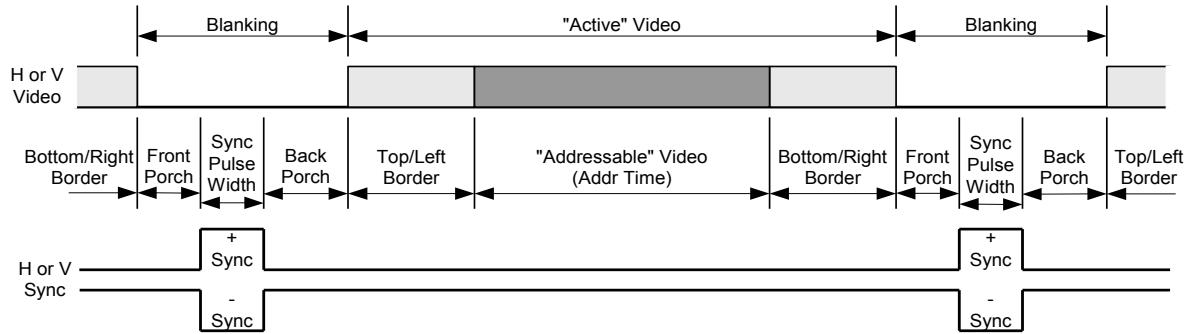


Figure 4-1: Video Timing Parameter Definitions

4.4.2 Type II Timing – Detailed

Tag Code	04h		Revision	0		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	O	O	O	O	O
Restrictions	There are no restrictions on how many Type II Timing Data Blocks may be provided in a DisplayID structure. Type II Timing Data Blocks may be used in base section and any extensions.					

The Type II Detailed Timing Data Block provides a more compact form for conveying detailed timing information than is the case with the original 18-byte Detailed Timing (now Type I) used in EDID. The Type II block is also capable of providing multiple timings, up to a maximum of 27, in a single data block. However, to be declared in a Type II Timing Data Block, the display timing must meet certain requirements:

All horizontal parameters (total H. pixels, active H. pixels, and H. sync offset) must be capable of being expressed as a multiple of 8 pixels. The horizontal active pixel count also may not exceed 4,096 pixels. Most common PC and television pixel formats, esp. those based on the concept of square pixels, will meet this requirement.

The maximum horizontal blanking value is 128 (1024 pixels); while this restriction will not be difficult for most timings, please note that this implies a maximum blanking percentage of 20% for a 4,096 x N timing, which may be difficult for a “CRT-like” timing at such high pixel counts.

The maximum width of the horizontal sync pulse value is 16 (128 pixels); the maximum width for the vertical is 16 lines. Type II Detailed Timings do not include border information.

The full definition of the format of the Type II Detailed Timing Data Block Structure and Descriptors are given in Table 4-20 and Table 4-21:

Table 4-19: Type II Detailed Timing Data Block

Offset	Value								Description/Format/Priority	
00 _h	04 _h								TYPE II TIMING – DETAILED	
01 _h	7	6	5	4	3	2	1	0	BLOCK Revision and Other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’ VALUES 0 → 7	
	0	0	0	0	0	—	—	—	RESERVED	
02 _h	B _h 16 _h 21 _h ...								Number of Payload Bytes in BLOCK = (N × 11) All Other Values RESERVED 11 → 242 1 ≤ N ≤ 22	
03 _h	11 BYTE DESCRIPTOR								1 st Type II ‘DETAILED’ TIMING	
0C _h	11 BYTE DESCRIPTOR								2 nd Type II ‘DETAILED’ TIMING ... (if present)	
										“ 2

Table 4-20: Type II Detailed Timing Descriptor

Byte	Value								Description / Format	
0, 1, 2	00 00 00 _h → FF FF FF _h								Pixel Clock + 10,000 0.01 → 167,772.16 Mega Pixels per Sec	
0	00 _h → FF _h								Low bits 7 → 0	
1	00 _h → FF _h								Middle bits 15 → 8	
2	00 _h → FF _h								High bits 23 → 16	
3	7	6	5	4	3	2	1	0	Timing Options	FLAGS
	1	—	—	—	—	—	—	—	PREFERRED ‘Detailed’ Timing	
	6	5	—	—	—	—	—	—	3D Stereo Support	FLAG
	0	0	—	—	—	—	—	—	This timing is always displayed monoscopic (no stereo)	
	0	1	—	—	—	—	—	—	This timing is always displayed in stereo	
	—	1	0	—	—	—	—	—	This timing is displayed in mono or stereo depending on a user action (wearing the stereo glasses, etc.)	
	—	1	1	—	—	—	—	—	RESERVED – do not use	
	—	—	4	—	—	—	—	—	Interface Frame Scanning Type	FLAG
	—	—	0	—	—	—	—	—	Progressive Scan Frame	
	—	—	1	—	—	—	—	—	Interlaced Scan Frame	
	—	—	—	3	—	—	—	—	Horizontal Sync Polarity	FLAG
	—	—	—	0	—	—	—	—	Negative Sync Polarity	
	—	—	—	1	—	—	—	—	Positive Sync Polarity	
	—	—	—	—	2	—	—	—	Vertical Sync Polarity	FLAG
	—	—	—	—	0	—	—	—	Negative Sync Polarity	
	—	—	—	—	1	—	—	—	Positive Sync Polarity	
	—	—	—	—	—	1	0	—	Reserved Bit	
	—	—	—	—	—	0	0	—	Set to 0	
4, 5	0 00 _h → 1 FF _h								Horizontal Active Image, (Pixels/8) - 1	1 → 512 Char
4	00 _h → FF _h								Low bits 7 → 0	
5	7	6	5	4	3	2	1	0	Horizontal Active Image (High bit) / Horizontal Blank	
	—	—	—	—	—	—	P	—	High bit 8 :	
	—	—	—	00 _h → 7F _h	—	—	—	—	Horizontal Blank , (Pixels/8) - 1	1 → 128 Char
6	MN _h								Horizontal Sync Front Porch/Offset & Width, (Pixels/8) - 1	
	M _h	—	—	—	—	—	—	—	Horizontal Sync Offset: bits 7 → 4	1 → 16 Char
	—	N _h	—	—	—	—	—	—	Horizontal Sync Width: bits 3 → 0	1 → 16 Char
7, 8	000 _h → FFF _h								Vertical Active Image Lines	1 → 4096 Lines
7	00 _h → FF _h								Low bits 7 → 0	
8	7	6	5	4	3	2	1	0	Vertical Active Image (High Bits) / Reserved	
	—	—	—	0 _h → F _h	—	—	—	—	High bits 11 → 8	
	—	—	0 _h	—	—	—	—	—	RESERVED Set to 0 _h	
9	00 _h → FF _h								Vertical Blank Lines	1 → 256 Lines

10	UV_h	Vertical Sync Front Porch/Offset and Width
	U _h	Vertical Sync Offset: bits 7 → 4 1 → 16 Lines
	V _h	Vertical Sync Width: bits 3 → 0 1 → 16 Lines

4.4.3 Type III Timing – Short

Tag Code	05 _h		Revision	0, 1		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	O	O	O	O	O
Restrictions	There are no restrictions on how many TYPE III TIMING DATA BLOCKS may be provided in a DISPLAYID structure. TYPE III TIMING DATA BLOCKS may be used in the BASE SECTION and any EXTENSIONS.					

The Type III Short Timing Data Block provides a means for supporting non-standard timings not documented in a specific VESA, CEA, or other discrete timing standards which are produced via one of the standard timing-generation algorithms such as the VESA CVT Standard. Priority of timings is in the order listed.

The Type III Short Timing Descriptor defined in Table 4-22 is not compliant with the 3 Byte CVT Codes defined in VESA's CVT and E-EDID Standards. The 3 byte CVT codes listed in the VESA DMT Standard shall not be used in Table 4-23.

The full definition of the format of the Type III Short Timing Data Block Structure and Descriptors are given in Table 4-21 and Table 4-22:

Table 4-21: Type III Short Timing Data Block

Offset	Value	Description/Format/Priority
00 _h	05 _h	TYPE III ‘SHORT’ TIMING TAG
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data
	— — — — — 0 0 1	REVISION ‘1’ VALUES 0 → 1
	0 0 0 0 0 — — —	RESERVED
02 _h	3 _h 6 _h 9 _h ...	Number of Payload Bytes in BLOCK = (N × 3) 3 → 246 All Other Values RESERVED 1 ≤ N ≤ 82
03 _h	3 BYTE DESCRIPTOR	1 st Type III ‘SHORT’ TIMING PRIORITY 1
06 _h	3 BYTE DESCRIPTOR	2 nd Type III ‘SHORT’ TIMING ... (if present) “ 2
.	.	.

Table 4-22: Type III Short Timing Descriptor

Byte	Value	Description/Format
0	7 6 5 4 3 2 1 0	Timing Options
	1 — — — — — — —	PREFERRED TIMING FLAG
	6 5 4 — — — — —	Timing Formula/algorithm TAG
	— 0 0 0 — — — —	VESA CVT, standard blanking
	0 0 1 — — — — —	VESA CVT, reduced blanking
	— — — — — — — —	All Other Values RESERVED
	— — — 3 2 1 0 —	Aspect Ratio TAG
	— — — 0 0 0 0 —	1: 1
	— — — 0 0 0 1 —	5: 4
	— — — 0 0 1 0 —	4: 3

				0	0	1	1	15: 9
				0	1	0	0	16: 9
				0	1	0	1	16: 10
				0	1	1	0	64: 27
				0	1	1	1	256: 135
				1	0	0	0	Not Defined
								All Other Values RESERVED
1	00_h → FF_h							Horizontal Active Image, (Pixels/8) - 1: 1 → 256 Char
								Horizontal Active Image
2	7	6	5	4	3	2	1	0
	0							Frame Transfer Type and Rate
	1							Progressive Timing
								Field Interlaced Timing
								FLAG
	6	5	4	3	2	1	0	Transfer Rate
								00 _h → 7F _h Frame/Field Refresh Rate 1 → 128Hz

4.4.4 Type IV Timing – DMT ID Code

Tag Code	06 _h		Revision	0			
Block Size	Variable						
Type Identifier	1	2	3	4	5	6	
Required (R) Optional (O)	O	O	O	O	O	O	
Restrictions	Timings cannot be declared as PREFERRED.						

Version 1 of the Type IV Short Timing Data Block provides a means for indicating support for DMT timings (VESA DMT Rev.13), CEA VIC Timings and HDMI VIC Timings. Priority of the timings is in the order listed. A DisplayID section can have multiple Type IV Timing data blocks. Priority is based on order they are listed in the section.

Appendix D shows an example of how the data block can be used to expose different type of timings.

Full definition of the format of the Type IV Short Timing Data Block Structure is given in Table 4-23.

Table 4-23: Type IV Short Timing Data Block

Offset	Value								Description/Format/Priority	
00 _h	06 _h								TYPE IV ‘SHORT’ TIMING	
01 _h	7	6	5	4	3	2	1	0	TAG	
	—	—	—	—	—	2	1	0	BLOCK Revision and Other Data	
	—	—	—	—	—	0	0	1	REVISION	
	—	—	—	—	—	—	—	—	VALUES 0 → 7	
	7	6	—	—	—	—	—	—	REVISION ‘1’	
	0	0	—	—	—	—	—	—	Timing Code Type	
	0	1	—	—	—	—	—	—	VALUES 0 → 3	
	1	0	—	—	—	—	—	—	DMT Timing Code	
	1	1	—	—	—	—	—	—	CEA VIC Timing Code	
	—	—	5	4	3	—	—	—	HDMI VIC Timing Code	
	—	—	0	0	0	—	—	—	Reserved Timing Code Type	
	—	—	—	—	—	—	—	—	RESERVED	
	—	—	—	—	—	—	—	—	RESERVED	
02 _h	01 _h → F8 _h								Number of Payload Bytes in BLOCK	
	All Other Values RESERVED								1 → 248	
03 _h	1 Byte Descriptor				1-byte Timing Code			Priority 1	TAG	
04 _h	1 Byte Descriptor				1-byte Timing Code			Priority 2	TAG	
	.								.	

4.4.5 Type V Timing – Short Descriptor

Tag Code	11_h		Revision	0	
Block Size	Variable				
Type Identifier	1	2	3	4	5
Required (R) Optional (O)	O	O	O	O	O
Restrictions	There are no restrictions on how many TYPE V TIMING DATA BLOCKS may be provided in a DISPLAYID structure. TYPE V TIMING DATA BLOCKS may be used in the BASE SECTION and any EXTENSIONS.				

The Type V Short Timing Data Block provides a means for supporting non-standard timings not documented in a specific VESA, CEA, or other discrete timing standards which are produced via one of the standard timing-generation algorithms such as VESA CVT Standard. Priority of the timings is in the order listed.

The full definition of the format of the Type V Short Timing Data Block Structure and Descriptors are given in Tables 4-25 and 4-26.

Table 4-245: Type V Short Timing Data Block

Offset	Value	Description / Format / Priority
00_h	11_h	TYPE V ‘SHORT’ TIMING TAG
01_h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data
	— — — — — 0 0 0	REVISION ‘0’ VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED
02_h	7_h E_h 15_h ...	Number of Payload Bytes in BLOCK = (N × 7) 7 → 245 All Other Values Reserved 1 ≤ N ≤ 53
03_h	7 BYTE DESCRIPTOR	1st Type V ‘SHORT’ TIMING DESCRIPTOR PRIORITY 1
0A_h	7 BYTE DESCRIPTOR	2nd Type V ‘SHORT’ TIMING DESCRIPTOR ... (if present) “ 2
.	.	.
.	.	.

Table 4-256: Type V Short Timing Descriptor

Byte	Value	Description / Format
0	7 6 5 4 3 2 1 0	Timing Options
	1 — — — — — — —	PREFERRED TIMING FLAG
	0 — — — — — — —	NON-PREFERRED TIMING FLAG
	6 5 — — — — — —	3D Stereo Support FLAG
	0 0 — — — — — —	This timing is always displayed monoscopic (no stereo)
	0 1 — — — — — —	This timing is always displayed in stereo
	1 0 — — — — — —	This timing is displayed in mono or stereo depending on a user action (wearing the stereo glasses, etc.)
	1 1 — — — — — —	RESERVED
	— — — — — 4 — —	
	— — — — — 1 — —	REFRESH RATE*(1000/1001) supported FLAG
	— — — — — 0 — —	REFRESH RATE*(1000/1001) not supported FLAG
	— — — — — — 1 0	Timing Formula/algorith TAG
	— — — — — — — 0	VESA CVT, reduced blanking v2
	— — — — — — — 1	VESA CVT, custom reduced blanking
	— — — — — — — 2_h → 3_h	All Other Values Reserved

	0	0	RESERVED						
2, 3	00 00_h → FF FF_h			Horizontal Active Image Pixels					
2	00 _h → FF _h			Low bits 7 → 0					
3	00 _h → FF _h			High bits 15 → 8					
4, 5	00 00_h → FF FF_h			Vertical Active Image Lines					
4	00 _h → FF _h			Low bits 7 → 0					
5	00 _h → FF _h			High bits 15 → 8					
6	7	6	5	4	3	2	1	0	Refresh Rate
	00 _h → FF _h			REFRESH RATE					
				1 → 256 Hz					

4.4.6 Type VI Timing – Detailed Descriptor

Tag Code	13_h		Revision	0
Block Size	Variable			
Type Identifier	1	2	3	4
Required (R)	O	O	O	O
Optional (O)				
Restrictions	There are no restrictions on how many TYPE VI TIMING DATA BLOCKS may be provided in a DISPLAYID structure. TYPE VI TIMING DATA BLOCKS may be used in the BASE SECTION and any EXTENSIONS.			

The Type VI Detailed Timing Data Block provides a method to expose higher precision pixel clock and high resolution detailed timings along with aspect ratio and vertical image size. It uses either 14 bytes or 17 bytes depending whether aspect ratio and image size information is exposed.

The full definition of the format of the Type VI Short Timing Data Block Structure and Descriptors are given in Tables 4-27 and 4-28.

Table 4-27: Type VI ‘Detailed’ Timing Data Block

Offset	Value	Description/Format/Priority
00 _h	13 _h	TYPE VI TIMING – DETAILED TAG
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data
	— — — — 0 0 0	REVISION ‘0’ VALUES 0
	0 0 0 0 0 — —	RESERVED
02 _h	...	Number of Payload Bytes in BLOCK = (N × 17 + M × 14) Where N represents timing with Aspect and Size information and M represents timing with no Aspect and Size information.
03 _h	(17 _h or 14 _h) BYTE DESCRIPTOR	1 st Type VI ‘DETAILED’ TIMING PRIORITY 1
3 + (17 _h or 14 _h)	(17 _h or 14 _h) BYTE DESCRIPTOR	2 nd Type VI ‘DETAILED’ TIMING ... (if present) 2
.	.	.

Table 4-28: Type VI Detailed Timing Descriptor

Bytes	Value	Description/format
0, 1, 2	00 00 00 _h → 1F FF FF _h	Pixel clock + 1,000 0.001 → 4,194.303 Mega Pixels per Sec
0	00 _h → FF _h	Low bits 7 → 0

1	$00_h \rightarrow FF_h$								Middle bits 15 → 8
2	7	6	5	4	3	2	1	0	Pixel Clocks (High Bits) & Timing Flags
	$0_h \rightarrow 3F_h$								High bits 21 → 16
	7								Timing Options
	1								PREFERRED ‘Detailed’ Timing
	0								NON-PREFERRED ‘Detailed’ Timing
	6								Aspect & Size Information
	-	0	-	-	-	-	-	-	Aspect & Size Information not included as part of the block, block ends at byte 13.
	-	1	-	-	-	-	-	-	Aspect & Size Information included as part of the block bytes 14 to 16 shall be populated as part of the block.
									It is recommended that Aspect & Size information be populated for preferred timing and stereo display timings.
3, 4	$00\ 00_h \rightarrow 3F\ FF_h$								Horizontal Active Image Pixels 1 → 16,384 Pixels
3	$00_h \rightarrow FF_h$								Low bits 7 → 0
4	7	6	5	4	3	2	1	0	Horizontal Active Image Pixels (High Bits) & Timing Flags
	$0_h \rightarrow 3F_h$								High bits 13 → 8
	7								Horizontal Sync Polarity
	0								Negative
	1								Positive
	6								Reserved Bit
	-	0							Set to 0
5, 6	$00\ 00_h \rightarrow 3F\ FF_h$								Vertical Active Image Lines 1 → 16,384 Lines
5	$00_h \rightarrow FF_h$								Vertical Active Image Lines [Low bits 7 → 0]
6	7	6	5	4	3	2	1	0	Vertical Active Image Lines (High Bits) & Timing Flags
	$0_h \rightarrow 3F_h$								Vertical Active Image Lines [High bits 13 → 8]
	7								Vertical Sync Polarity
	0								Negative
	1								Positive
	6								Reserved Bit
	-	0							Set to 0
7, 9	$00\ 00_h \rightarrow F\ FF_h$								Horizontal Blank Pixels 1 → 4,096 Pixels
7	$00_h \rightarrow FF_h$								Horizontal Blank Pixels [Low bits 7 → 0]
8, 9	$00\ 00_h \rightarrow F\ FF_h$								Horizontal Offset (Horizontal Front Porch) 1 → 4,096 Pixels
8	$00_h \rightarrow FF_h$								Horizontal Offset [Low bits 7 → 0]
9	7	6	5	4	3	2	1	0	Horizontal Blank Pixels (High Bits) & Horizontal Offset (High Bits)
	$0_h \rightarrow F_h$								Horizontal Blank Pixels [High bits 11 → 8]
	$0_h \rightarrow F_h$								Horizontal Offset [High bits 11 → 8]
10	$00_h \rightarrow FF_h$								Horizontal Sync Width 1 → 256 Pixels
	$00_h \rightarrow FF_h$								Horizontal Sync Width
11	$00_h \rightarrow FF_h$								Vertical Blank Lines 1 → 256 Lines
	$00_h \rightarrow FF_h$								Vertical Blank Lines
12	$00_h \rightarrow FF_h$								Vertical Offset (Vertical Front Porch) 1 → 256 Lines
	$00_h \rightarrow FF_h$								Vertical Offset (Vertical Front Porch)
13	7	6	5	4	3	2	1	0	Vertical Sync Width & Timing Flags
	$0_h \rightarrow F_h$								Vertical Sync Width 1 → 16
	7								Interface Frame Scanning Type
	0								Progressive Scan Frame
	1								Interlaced Scan Frame
	6	5							3D Stereo Support FLAG
	-	0	0						This timing is always displayed monoscopic (no stereo)

	0	1							This timing is always displayed in stereo
	-	1	0	-	-	-	-	-	This timing is displayed in mono or stereo depending on a user action (wearing the stereo glasses, etc.)
		1	1						RESERVED
			0						Reserved Bit
			0						Set to 0
14	00_h → FF_h						Aspect Multiplier	0 → 255	
	00 _h → FF _h						Aspect Multiplier	Aspect Ratio = Aspect Multiplier x 3 / 256 0 → 2.99	
15	00 00_h → F FF_h						Vertical Image Base Size	1 → 4,096 mm	
	00 _h → FF _h						Vertical Image Size = Vertical Image Base Size x Size Multiplier		
	00 _h → FF _h						Vertical Image Base Size [Low bits 7 → 0]		
16	7	6	5	4	3	2	1	0	Vertical Image Base Size (High Bits) & Size Multiplier
									Vertical Image Base Size [High bits 11 → 8]
	0 _h → F _h						Size Multiplier	1 → 16	
							If Vertical Image Base Size = 2,550 mm & Size Multiplier = 1 => Vertical Image Size = 2,550 mm		
							If Vertical Image Base Size = 2,550 mm & Size Multiplier = 16 => Vertical Image Size = 40,800 mm		

4.5 Supported Standard Timing Modes

The Supported Video Timing Modes Data Block consists of a single field of one-bit flags used to indicate support for VESA and other common timing standards in a very compact form. Any timing not covered by these flags may be described using one of the Video Timing Mode Data Blocks or other timing-related blocks. If the bit is not set, it does not mean that timing is not supported; timing may be supported using other data blocks.

Table 4-29: Supported Video Timing Modes Data Block

Offset	Value/Bit	Description
00 _h	07_h → 08_h	SUPPORTED STANDARD TIMING MODES TAG
	07 _h	VESA TIMING STANDARD Section 4.5.1
	08 _h	CEA TIMING STANDARD Section 4.5.2
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data
	— — — — — 0 0 0	REVISION ‘0’ VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED
02 _h	0N _h	Number of Payload Bytes in Block based on Tag
03 _h	FLAG BITS	1 st VIDEO TIMING MODE GROUP
04 _h	FLAG BITS	2 nd VIDEO TIMING MODE GROUP
09 _h	FLAG BITS	9 th VIDEO TIMING MODE GROUP
0A _h	FLAG BITS	10 th VIDEO TIMING MODE GROUP

4.5.1 VESA Timing Standard

Tag Code	07 _h	Revision	0
Block Size	Fixed		
Type Identifier	1	2	3
Required (R) Optional (O)	O	O	O
		4	5
		6	O

Restrictions	No more than one Supported Video Timing Modes Data Block of the same TAG code may be provided in any given DisplayID structure. Timings cannot be declared as preferred.
---------------------	--

The VESA Timings Standard Data Block cannot be used for defining the priority order of the listed timing modes. It can only be used to define support for the listed timing mode.

Bits set to “1” within Table 4-30 indicate support for that listed VESA timing as a Factory Supported Mode which are defined as those timing modes that result in an video image on the display’s screen, properly sized and centered, as the product is delivered from the supplier. This does not cover timings which might be later created and/or stored as “user-defined” modes, etc.

Refer to VESA DMT, Rev. 13 for timing parameter definitions of DMTs listed in Table 4-30.

Table 4-30: Supported VESA Timings Standard Data Block

Offset	Value/Bit	Description	Notes or Source of Controlling Authority
00_h	07_h	TYPE I VESA DMT TIMINGS	TAG
01_h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	0 0 0 0 0 0 0 0	REVISION ‘0’	VALUES 0 → 7
	0 0 0 0 0 0 0 0	RESERVED	FLAGS / TAG
02_h	0A_h	Number of Payload Bytes in BLOCK	
03_h	1	640 x 350 @ 85Hz	VESA DMT
	1	640 x 400 @ 85Hz	VESA DMT
	1	720 x 400 @ 85Hz	VESA DMT
	1	640 x 480 @ 60Hz	VESA DMT
	1	640 x 480 @ 72Hz	VESA DMT
	1	640 x 480 @ 75Hz	VESA DMT
	1	640 x 480 @ 85Hz	VESA DMT
	1	800 x 600 @ 56Hz	VESA DMT
	1	800 x 600 @ 60Hz	VESA DMT
	1	800 x 600 @ 72Hz	VESA DMT
04_h	1	800 x 600 @ 75Hz	VESA DMT
	1	800 x 600 @ 85Hz	VESA DMT
	1	800 x 600 @ 120Hz (RB)	VESA DMT
	1	848 x 480 @ 60Hz	VESA DMT
	1	1024 x 768 @ 43Hz (INT)	VESA DMT
	1	1024 x 768 @ 60Hz	VESA DMT
	1	1024 x 768 @ 70Hz	VESA DMT
	1	1024 x 768 @ 75Hz	VESA DMT
	1	1024 x 768 @ 85Hz	VESA DMT
	1	1024 x 768 @ 120Hz (RB)	VESA DMT
05_h	1	1152 x 864 @ 75Hz	VESA DMT
	1	1280 x 768 @ 60Hz (RB)	VESA DMT
	1	1280 x 768 @ 60Hz	VESA DMT
	1	1280 x 768 @ 75Hz	VESA DMT
	1	1280 x 768 @ 85Hz	VESA DMT
	1	1280 x 768 @ 120Hz (RB)	VESA DMT
	1	1280 x 800 @ 60Hz (RB)	VESA DMT
	1	1280 x 800 @ 60Hz	VESA DMT
	1	1280 x 800 @ 85Hz	VESA DMT
	1	1280 x 800 @ 120Hz (RB)	VESA DMT
06_h	1	1280 x 800 @ 60Hz	VESA DMT

		1			1280 x 800 @ 75Hz	VESA DMT
		1			1280 x 800 @ 85Hz	VESA DMT
		1			1280 x 800 @ 120Hz (RB)	VESA DMT
		1			1280 x 960 @ 60Hz	VESA DMT
07 _h				1	1280 x 960 @ 85Hz	VESA DMT
				1	1280 x 960 @ 120Hz (RB)	VESA DMT
			1		1280 x 1024 @ 60Hz	VESA DMT
			1		1280 x 1024 @ 75Hz	VESA DMT
			1		1280 x 1024 @ 85Hz	VESA DMT
		1			1280 x 1024 @ 120Hz (RB)	VESA DMT
	1				1360 x 768 @ 60Hz	VESA DMT
	1				1360 x 768 @ 120Hz (RB)	VESA DMT
08 _h				1	1400 x 1050 @ 60Hz (RB)	VESA DMT
				1	1400 x 1050 @ 60Hz	VESA DMT
			1		1400 x 1050 @ 75Hz	VESA DMT
			1		1400 x 1050 @ 85Hz	VESA DMT
			1		1400 x 1050 @ 120Hz (RB)	VESA DMT
		1			1440 x 900 @ 60Hz (RB)	VESA DMT
	1				1440 x 900 @ 60Hz	VESA DMT
	1				1440 x 900 @ 75Hz	VESA DMT
09 _h				1	1440 x 900 @ 85Hz	VESA DMT
				1	1440 x 900 @ 120Hz (RB)	VESA DMT
			1		1600 x 1200 @ 60Hz	VESA DMT
			1		1600 x 1200 @ 65Hz	VESA DMT
			1		1600 x 1200 @ 70Hz	VESA DMT
		1			1600 x 1200 @ 75Hz	VESA DMT
	1				1600 x 1200 @ 85Hz	VESA DMT
	1				1600 x 1200 @ 120Hz (RB)	VESA DMT
0Ah				1	1680 x 1050 @ 60Hz (RB)	VESA DMT
				1	1680 x 1050 @ 60Hz	VESA DMT
			1		1680 x 1050 @ 75Hz	VESA DMT
			1		1680 x 1050 @ 85Hz	VESA DMT
			1		1680 x 1050 @ 120Hz (RB)	VESA DMT
		1			1792 x 1344 @ 60Hz	VESA DMT
	1				1792 x 1344 @ 75Hz	VESA DMT
	1				1792 x 1344 @ 120Hz (RB)	VESA DMT
0B _h				1	1856 x 1392 @ 60Hz	VESA DMT
				1	1856 x 1392 @ 75Hz	VESA DMT
			1		1856 x 1392 @ 120Hz (RB)	VESA DMT
			1		1920 x 1200 @ 60Hz (RB)	VESA DMT
			1		1920 x 1200 @ 60Hz	VESA DMT
		1			1920 x 1200 @ 75Hz	VESA DMT
	1				1920 x 1200 @ 85Hz	VESA DMT
	1				1920 x 1200 @ 120Hz (RB)	VESA DMT
0C _h				1	1920 x 1440 @ 60Hz	VESA DMT
				1	1920 x 1440 @ 75Hz	VESA DMT
			1		1920 x 1440 @ 120Hz (RB)	VESA DMT
		1			2560 x 1600 @ 60Hz (RB)	VESA DMT

	1				2560 x 1600 @ 60Hz	VESA DMT
	1				2560 x 1600 @ 75Hz	VESA DMT
	1				2560 x 1600 @ 85Hz	VESA DMT
	1				2560 x 1600 @ 120Hz (RB)	VESA DMT

Notes:

1. It is not possible to identify any timing as preferred within this block. Identifying a preferred timing requires the use of one of the Video Timing Mode Data Blocks.
2. The 1-bit flags of this section shall not be used to determine the range of format or refresh rate support, or any other timing range limits of the display. Any 1-bit flag not set in this section does not preclude that timing from being supported via the timing range limits of the display, but only that it is not a factory supported mode.
3. All timings listed in Table 4-30 are “Progressive” scan, unless labeled with “INT”. “INT” is an “Interlaced” scan. Interlaced refers to conventional 2:1 interlaced formats.
4. Timings labeled “RB” are “Reduced Blanking”. For more information on Reduced Blanking Timings, refer to the VESA CVT Standard.
5. Bits set to “1” within this section indicate support for that listed timing as a Factory Supported Mode which is defined as those timing modes that result in a video image on the display’s screen, properly sized and centered, as the product is delivered from the supplier. This does not cover timings which might be later created and/or stored as “user-defined” modes, etc.).

4.5.2 CEA Timing Standard

Tag Code	08 _h		Revision	0		
Block Size	Fixed					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	O	O	O	O	O
Restrictions	No more than one Supported Video Timing Modes Data Block of the same tag code may be provided in any given DisplayID structure. Timings cannot be declared as preferred.					

Supported CEA Timing Standard Data Blocks cannot be used for defining the priority order of listed timing modes. It can only be used to define support for the listed timing mode.

Bits set to “1” within Table 4-27 indicate support for that listed CEA timing as a Factory Supported Mode. Factory Supported Modes are defined as timing modes that result in a video image on the display’s screen, properly sized and centered, as the product is delivered from the supplier.

Refer to the CEA-861- (most recent version), - A DTV Profile for Uncompressed High Speed Digital Interfaces Standard for the timing parameter definitions of the CEA DTV Timings listed in Table 4-31.

Note: The term 60Hz support as used in this section also includes support for 59.94Hz (or 60Hz * 1000/1001) timings where appropriate.

Table 4-31: Supported CEA Timing Standard Data Block

Offset	Value								Description	
00h	08h								Type II CEA DTV	Tag
01h	7	6	5	4	3	2	1	0	BLOCK Revision and other Notes	
					0	0	0	0	REVISION ‘0’	

	0	0	0	0	0		RESERVED	FLAGS / TAG
02_h	08_h					Number of Payload Bytes in BLOCK		
						H x V Pixel Formats	Vertical Refresh Rate	Picture Aspect Ratio (H:V)
03_h					1	640 x 480p	59.94/60Hz	4:3
					1	720 x 480p	59.94/60Hz	4:3
				1		720 x 480p	59.94/60Hz	16:9
			1			1280 x 720p	59.94/60Hz	16:9
			1			1920 x 1080i	59.94/60Hz	16:9
		1				720(1440) x 480i	59.94/60Hz	4:3
	1					720(1440) x 480i	59.94/60Hz	16:9
	1					720(1440) x 240p	59.94/60Hz	4:3
04_h					1	720(1440) x 240p	59.94/60Hz	16:9
					1	2880 x 480i	59.94/60Hz	4:3
				1		2880 x 480i	59.94/60Hz	16:9
			1			2880 x 240p	59.94/60Hz	4:3
			1			2880 x 240p	59.94/60Hz	16:9
			1			1440 x 480p	59.94/60Hz	4:3
		1				1440 x 480p	59.94/60Hz	16:9
	1					1920 x 1080p	59.94/60Hz	16:9
05_h					1	720 x 576p	50Hz	4:3
					1	720 x 576p	50Hz	16:9
				1		1280 x 720p	50Hz	16:9
			1			1920 x 1080i	50Hz	16:9
			1			720(1440) x 576i	50Hz	4:3
		1				720(1440) x 576i	50Hz	16:9
	1					720(1440) x 288p	50Hz	4:3
	1					720(1440) x 288p	50Hz	16:9
06_h					1	2880 x 576i	50Hz	4:3
					1	2880 x 576i	50Hz	16:9
				1		2880 x 288p	50Hz	4:3
			1			2880 x 288p	50Hz	16:9
			1			1440 x 576p	50Hz	4:3
		1				1440 x 576p	50Hz	16:9
	1					1920 x 1080p	50Hz	16:9
	1					1920 x 1080p	23.97/24Hz	16:9
07_h					1	1920 x 1080p	25Hz	16:9
					1	1920 x 1080p	29.97/30Hz	16:9
				1		2880 x 480p	59.94/60Hz	4:3
			1			2880 x 480p	59.94/60Hz	16:9
			1			2880 x 576p	50Hz	4:3
		1				2880 x 576p	50Hz	16:9
	1					1920 x 1080i	50Hz	16:9
	1					1920 x 1080i	100Hz	16:9
08_h					1	1280 x 720p	100Hz	16:9
					1	720 x 576p	100Hz	4:3
				1		720 x 576p	100Hz	16:9
			1			720(1440) x 576i	100Hz	4:3
		1				720(1440) x 576i	100Hz	16:9
	1					1920 x 1080i	119.88/120Hz	16:9

	1					1280 x 720p	119.88/120Hz	16:9
	1					720 x 480p	119.88/120Hz	4:3
09 _h					1	720 x 480p	119.88/120Hz	16:9
					1	720(1440) x 480i	119.88/120Hz	4:3
					1	720(1440) x 480i	119.88/120Hz	16:9
					1	720 x 576p	200Hz	4:3
					1	720 x 576p	200Hz	16:9
					1	720(1440) x 576i	200Hz	4:3
					1	720(1440) x 576i	200Hz	16:9
					1	720 x 480p	239.76/240Hz	4:3
0A _h					1	720 x 480p	239.76/240Hz	16:9
					1	720(1440) x 480i	239.76/240Hz	4:3
					1	720(1440) x 480i	239.76/240Hz	16:9
					1	1280 x 720p	23.97/24Hz	16:9
					1	1280 x 720p	25Hz	16:9
					1	1280 x 720p	29.97/30Hz	16:9
					1	1920x1080p	120Hz	16:9
					1	1920x1080p	100Hz	16:9

Notes:

1. All H x V Pixel Formats listed in Table 4-31 that include the suffix “p” are “Progressive” scan and all H x V Pixel Formats that include the suffix “i” are “Interlaced” scan. Interlaced refers to conventional 2:1 interlaced formats.
2. The one-bit flags of this section shall not be used to determine the range of format or refresh rate support, or any other timing range limits of the display. Any one-bit flag not set in this section does not preclude that timing from being supported via the timing range limits of the display but only that it is not a factory supported mode.

4.6 Video Timing Range Limits

Tag Code	09 _h			Revision	0		
Block Size	Fixed						
Type Identifier	1	2	3	4	5	6	
Required (R) Optional (O)	O	O	O	O	O	O	
Restrictions	Multiple Video Timing Range Limits Data Blocks may appear in any DisplayID structure, and used if the display supports operation over two or more non-contiguous ranges of timings.						

The Video Timing Range Limit Block is used to convey the supported range of vertical and horizontal frequencies to a source along with the maximum supported pixel clock. For a continuous frequency device, any timing that lies within the specified range will ensure a displayable image. The picture may not be properly sized or centered, as guaranteed with explicitly reported timing. However, the display should synchronize to the timing.

Table 4-262: Video Timing Range Limits Data Block

Offset	Value								Description/Format	
00 _h	09 _h								VIDEO TIMING RANGE LIMITS DATA BLOCK	
01 _h	7	6	5	4	3	2	1	0	TAG	
	—	—	—	—	—	0	0	0	BLOCK Revision and other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’	VALUE 0 → 7
	0	0	0	0	0	—	—	—	RESERVED BITS	

02_h	0F_h	Number of Payload Bytes in BLOCK All Other Values RESERVED							15
05_h 04_h 03_h	00 00 00_h → FF FF FF_h	Minimum pixel clock ÷ 10,000							0.01 → 167,772.16 MHz
08_h 07_h 06_h	00 00 00_h → FF FF FF_h	Maximum pixel clock ÷ 10,000							0.01 → 167,772.16 MHz
09_h	00_h → FF_h	Minimum horizontal frequency							0 → 255 KHz
0A_h	00_h → FF_h	Maximum horizontal frequency							0 → 255 KHz
0C_h 0B_h	00 00_h → FF FF_h	Minimum horizontal blanking							0 → 65,535 Pixels
0D_h	00_h → FF_h	Minimum vertical refresh (field/frame) rate							0 → 255Hz
0E_h	00_h → FF_h	Maximum vertical refresh (field/frame) rate							0 → 255Hz
10_h 0F_h	00 00_h → FF FF_h	Minimum vertical blanking							0 → 65,535 Lines
11_h	7 6 5 4 3 2 1 0	Video Timing Support over this range							Flag
	1	Supports interlaced operation							
	1	VESA CVT, standard blanking timings							
	1	VESA CVT, reduced-blanking timings							
	1	Discrete frequency display device							
		0 0 0 0 0 0 0							RESERVED

4.6.1 Minimum Pixel Clock

Minimum pixel clock is a required 3-byte field to specify display device supported minimum pixel clock frequency. It is specified in 10,000Hz units. It can be in the range of 0.01MHz to 167772.16MHz.

4.6.2 Maximum Pixel Clock

Maximum pixel clock is a required 3-byte field to specify display device supported maximum pixel clock frequency. It is specified in 10,000Hz units. It can be in the range 0.01MHz to 167772.16MHz.

4.6.3 Minimum Horizontal Frequency

This is a required field to specify minimum horizontal frequency of the display device. Horizontal frequency is defined as pixel_clock/vertical_total. It is specified in KHz units and can be up to 255 KHz.

4.6.4 Maximum Horizontal Frequency

This is a required field to specify maximum horizontal frequency of the display device. It is specified in KHz units and can be up to 255 KHz.

4.6.5 Minimum Horizontal Blanking

This is a required field to specify minimum horizontal blanking period of the display device. It is specified in number of pixels between horizontal active and horizontal total.

4.6.6 Minimum Vertical Frequency

This is a required field to specify minimum vertical frequency of the display device. Vertical frequency is defined as (pixel_clock/vertical_total)/vertical_total. It is specified in Hz and can be up to 255Hz.

4.6.7 Maximum Vertical Frequency

This is a required field to specify maximum vertical frequency of the display device. It is specified in Hz and can be up to 255Hz.

4.6.8 Minimum Vertical Blanking

This is a required field to specify minimum vertical blanking period of the display device. It is specified in number of lines between vertical active and vertical total.

4.7 Product Serial Number

Tag Code	0A_h		Revision	0		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R)	O	O	O	O	O	O
Restrictions	Only one Serial Number Data Block may appear in any given DisplayID structure.					

The Serial Number Data Block provides a means for conveying additional or extended product serial number information beyond the 32-bit binary value provided for in the Product Identification Data Block. It is permissible to provide a serial number within the Product Identification Data Block, and a separate Serial Number Data Block.

If a serial number is provided in either location, the value provided must be unique to that specific product; no other product of that model or type be produced which has the same value in a Serial Number Data Block in its DisplayID structure. The only exception is for multiple products which contain the value “00000000” in the Serial Number field of the Product Identification Data Block.

If the Serial Number Field of the Product Identification Data Block contains “00000000”, it is permissible for a unique serial number to be provided via a Serial Number Data Block. (See Section 4.1.3 for additional information.) Serial number information is provided in this block in the form of an ASCII string (Printable Characters only). An end null character is not required.

Table 4-33: Serial Number Data Block

Offset	Value								Description/Format	
00_h	0A_h								Product Serial Number	TAG
01_h	7 6 5 4 3 2 1 0								BLOCK Revision and other Data	
	0 0 0 0 0 0 0 0								REVISION ‘0’	VALUE 7 → 0
	0 0 0 0 0 0 0 0								RESERVED BITS	
02_h	00_h → F8_h								Number of Payload Bytes in BLOCK	N = 0 → 248
	All Other Values RESERVED									
03_h	Printable ASCII Character								1st character of SN string	
04_h	Printable ASCII Character								2nd character of SN string	
.	.								.	
.	.								.	
.	.								.	
(N - 1)_h	Printable ASCII Character								Final character of SN string	

4.8 General Purpose ASCII String

Tag Code	0B_h		Revision	0		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R)	O	O	O	O	O	O
Restrictions	There are no restrictions on how many General-Purpose ASCII Data Blocks may be provided within a given DisplayID structure. However, note that the ordering of these blocks may be important in some applications.					

The General-Purpose ASCII Data Block provides a means of conveying ASCII string data, one string per data block, as needed for some display systems or applications. Each string may be up to 248

characters long, to comply with the 251-byte limit on DisplayID section length. End null character is not required.

The format of the General-Purpose ASCII Data Block is identical to that of the Serial Number Data Block. The only difference is that the Serial Number Data Block is specifically reserved for carrying serial number strings only.

Table 4-274: General-Purpose ASCII String Data Block

Offset	Value								Description/Format	
00_h	0B_h								General Purpose ASCII String	
01_h	7	6	5	4	3	2	1	0	BLOCK Revision and other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’	
	0	0	0	0	0	—	—	—	RESERVED BITS	
02_h	00_h → F8_h								Number of Payload Bytes in BLOCK N = 0 → 248	
All Other Values RESERVED										
03_h	Printable ASCII Character								1st character of string	
04_h	Printable ASCII Character								2nd character of string	
.	.								.	
.	.								.	
.	.								.	
(N - 1) _h	Printable ASCII Character								Final character of string	

4.9 Display Device Data

Tag Code	0C_h		Revision	0		
Block Size	Fixed					
Type Identifier	1	2	3	4	5	6
Required (R)	O	R	O	O	O	O
Restrictions	No more than one Display Device data block may be provided in any DisplayID structure.					

The Device Data Block is intended to provide information concerning the characteristics of a basic display device (transducer); i.e., the characteristics of an LCD or PDP module, etc., as used within a monitor, notebook PC, or other such product. The Device Data Block is required for panels with a DisplayID structure intended for use in such “embedded” applications, “direct drive” monitors, and similar situations, and may be useful in other DisplayID structures as well.

Table 4-285: Display Device Data Block

Offset	Value								Description/Format	
00_h	0C_h								DISPLAY DEVICE DATA BLOCK	
01_h	7	6	5	4	3	2	1	0	BLOCK Revision and Other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’	
	0	0	0	0	0	—	—	—	RESERVED	
02_h	0D_h								Number of Payload Bytes in BLOCK	
03_h	DESCRIPTOR								Display Device Technology	
04_h	DESCRIPTOR								Device operating mode	
05_h → 08_h	DESCRIPTOR								Device native pixel format	
09_h → 0A_h	DESCRIPTOR								Aspect ratio and orientation	
0B_h	DESCRIPTOR								Sub-pixel layout/configuration/shape	
0C_h → 0D_h	DESCRIPTOR								Horizontal and vertical dot/pixel pitch	

0E_h	_DESCRIPTOR	Color bit depth
0F_h	_DESCRIPTOR	Response time

4.9.1 Display Device Technology

This byte describes the technology used by the display device, given as two four bit tags: one giving the class of technology (e.g., LCD, PDP, CRT, etc.) and the other giving the sub-type within that class (e.g., for LCDs, sub-types might include AM-TN, AM-IPS, AM-VA, etc.).

Main Technology Tag: The main class of display technology is described by upper four bits (7- 4) of this byte.

The meaning of the lower four bits of this byte varies depending on the main technology class indicated in upper four bits; these are considered the technology sub-type identifiers. The byte is allocated and these tags are defined in Table 4-36.

Table 4-296: Display Device Technology & Sub-type Codes

Offset	Value								Description/Format	
	7	6	5	4	3	2	1	0		
03 _h	Tech. type	Sub-type	Display Device Technology Sub-type Codes							
	0 _h		CRT TAG							
	0 _h		Monochrome CRT							
	1 _h		Standard tricolor CRT							
	2 _h		Other/undefined							
	3 _h → F _h		RESERVED							
	1 _h		LCD (all types) TAG							
	0 _h		Passive matrix TN (includes HTN, STN, etc.)							
	1 _h		Passive matrix cholesteric LC							
	2 _h		Passive matrix ferroelectric LC							
	3 _h		Other passive matrix LC type							
	4 _h		Active-matrix TN							
	5 _h		Active-matrix IPS (all types)							
	6 _h		Active-matrix VA (all types)							
	7 _h		Active-matrix OCB							
	8 _h		Active matrix ferroelectric							
	9 _h → E _h		RESERVED							
	F _h		Other LC type							
	2 _h		Plasma display (PDP) (all types) TAG							
	0 _h		DC plasma							
	1 _h		AC plasma							
	2 _h → F _h		RESERVED							
	3 _h		Electroluminescent, except OEL/OLED							
	4 _h		Inorganic LED							
	5 _h		Organic LED/OEL							
	6 _h		FED or sim. “cold-cathode,” phosphor-based types							
	7 _h		Electrophoretic							
	8 _h		Electrochromic							
	9 _h		Electromechanical							
	A _h		Electrowetting							
	B _h		RESERVED							
	C _h		RESERVED							
	D _h		RESERVED							
	E _h		RESERVED							
	F _h		Other type not defined here							

4.9.2 Display Device Operating Mode & Flags

The upper four bits of this byte identify the default or primary operating mode for the display device described by this data block, as defined by Table 4-37.

Table 4-307: Device Operating Modes Codes (upper four bits of byte 04h)

Offset	Value								Description/Format
	7	6	5	4	3	2	1	0	Bits
04h	0 _h	Lower 4 bits of byte 04h. (See Table 4-33)							Direct-view reflective display device relying on ambient lighting (i.e., no illumination source provided)
	1 _h								Direct-view reflective display device providing an illumination source but which in the default mode operates using ambient lighting.
	2 _h								Direct-view reflective display device providing an illumination source which is active by default.
	3 _h								Direct-view transmissive display device relying on ambient lighting (i.e., no illumination source provided)
	4 _h								Direct-view transmissive display device providing an illumination source but which in the default mode operates using ambient lighting.
	5 _h								Direct-view transmissive display device providing an illumination source which is active by default.
	6 _h								Direct-view emissive display device
	7 _h								Direct-view transflective display device which is normally used in the reflective mode (backlight off by default)
	8 _h								Direct-view transflective display device which is normally used in the transmissive mode (backlight on by default)
	9 _h								Transparent display whose image is seen by ambient light.
	A _h								Transparent emissive display
	B _h								Projection device using reflective light modulator (e.g. a Microelectromechanical systems (MEMs)-based projection engine)
	C _h								Projection device using transmissive light modulator (e.g., a conventional LCD-based projection engine)
	D _h								Projection device using emissive image transducer (e.g., a CRT projection engine)
	E _h								RESERVED; do not use
	F _h								RESERVED; do not use

The lower four bits of this byte are defined as flags providing additional information relating to the device operating mode as defined by Table 4-38.

Table 4-38: Device Operating Mode Flag Bits (lower four bits of byte 04h)

Offset	Value								Description/Format
	7	6	5	4	3	2	1	0	Bits
04h	Upper 4 bits of byte 04h. (See Table 4-32)		1	—	—	—	—	—	Bit 3. If set, this bit indicates that the device's backlight may be switched on and off using control method defined for the interface in use.
		—	1	—	—	—	—	—	Bit 2. If set, this bit indicates that the intensity of the device's backlight may be controlled used the method defined for the interface in use.
					0				RESERVED at 0.
						0			RESERVED at 0.

4.9.3 Device Native Pixel Format

These bytes describe the number of physical pixels in the horizontal and vertical directions, where those directions are defined relative to the local horizontal (i.e., floor, desktop, etc.) with the display in its normal or default orientation. (See Section 4.9.4 for the definition of bytes containing aspect ratio and orientation information.)

If all bytes in this section are set to 00h, it is assumed that the display does not have a fixed pixel format as is typical of a CRT or vector-scan display. This information should not be used to infer the physical aspect ratio or size of the display, as there is no requirement that the physical pixels enumerated by these bytes are themselves of a “square” (1:1) aspect ratio.

If the physical pixels of the display device are not arranged in an orthogonal/rectangular array (a “delta-pixel” type), these bytes shall contain the X and Y pixel counts of the native format of the display device, defining the maximum number of image pixels which can be fully resolved. A “checkerboard” pattern of this pixel count is resolvable by the device).

Table 4-39: Device Native Pixel Format Bytes

Offset	Value								Description/Format
	7	6	5	4	3	2	1	0	Bits
05h	0000h → FFFFh								Horizontal pixel count, bits 0→7
06h									Horizontal pixel count, bits 8→15
07h	0000h → FFFFh								Vertical pixel count, bits 0→7
08h									Vertical pixel count, bits 8→15

Note: Both the horizontal and vertical pixel counts are given as 16-bit values, to be interpreted as the pixel count on that axis, minus one; this provides a maximum permissible format of 65,536 x 65,536 pixels.

4.9.4 Physical Aspect Ratio and Orientation

These two bytes provide information on the physical aspect ratio (i.e., the ratio of physical size of long axis relative to shorter axis of image area of the display device) and default orientation of the display. This is not necessarily the ratio of the physical or logical pixel count in these two axes, as given in the previous section, as the pixels may be “non-square” (i.e., not of a 1:1 physical aspect ratio themselves).

The bytes are defined as shown in Table 4-40.

Table 4-40: Display Physical Aspect Ratio & Orientation

Offset	Value								Description/Format
	7	6	5	4	3	2	1	0	
09 _h	0_h → FF_h								Bits
0A _h									Aspect ratio; The aspect ratio is defined as the ratio of the physical dimensions of the image area of the display device, taken as long axis/short axis (and as such is a dimensionless quantity), expressed as a three-significant-figure value in the range of 1.00 to 3.55. The value stored here is given by: AR = (Long-axis image area size)/(Short-axis image area size) Stored value = (AR-1) * 100 For example, a display providing an image area of 160 mm x 90 mm (a 16:9, or 1.78:1, aspect ratio), would provide a value in this byte of 078 (decimal), or 4Eh. Note that this value is the same whether the “long side” of the image area is normally horizontal or normally vertical.
	7	6							Bits 7,6 – Default Orientation
	0	0							Landscape (long axis horizontal)
	0	1							Portrait (long axis vertical)
	1	0	–	–	–	–	–	–	Orientation not fixed (display may be rotated by the user; status information may need to be read “on the fly” to determine the current orientation.)
	1	1							Undefined.
		5	4						Bits 5,4 – Rotation Capability
		0	0						No rotation capability
	–	–	0	1	–	–	–	–	Display may be rotated 90 degrees clockwise from the default orientation described here.
	–	–	1	0	–	–	–	–	Display may be rotated 90 degrees counterclockwise from the default orientation described here.
	–	–	1	1	–	–	–	–	Display may be rotated 90 degrees in either direction from the default orientation described here.
				3	2				Bits 3,2 – “Zero pixel” Location The “zero pixel” is the screen location at which the scan begins for each new frame or field. This location is given as seen from the normal viewing position, with the display in its default orientation, per the following:
	–	–	–	–	0	0	–	–	Upper left corner
	–	–	–	–	0	1	–	–	Upper right corner
	–	–	–	–	1	0	–	–	Lower left corner
	–	–	–	–	1	1	–	–	Lower right corner
						1	0		Note that for a display which does not use a “raster scan” format, identifiable by the contents of Bits 1 and 0 (below), this information is irrelevant and the contents of these bits shall be ignored. Bits 1,0 – Scan Direction
	–	–	–	–	–	–	0	0	The scan direction for the display is not defined (e.g., a vector scan or other display type of indefinite scan direction & format).
	–	–	–	–	–	–	0	1	Fast (line) scan is along the long axis, slow (frame or field) scan is along the short axis.

	-	-	-	-	-	-	1	0	Fast (line) scan is along the short axis, slow (frame or field) scan is along the long axis.
	-	-	-	-	-	-	1	1	Undefined; this code is reserved and not to be used at this time.

4.9.5 Sub-pixel Information (Layout/configuration)

Sub-pixel layout and configuration is identified by a code number as defined in Table 4-37. This refers only to the sub-pixel physical layout and relative size of the features within each pixel; the actual physical size of the pixel is given in the next two bytes.

A diagram is provided in Appendix A which shows typical examples of the various defined layouts. This table is intended to be compatible with the code definitions used in VESA MCCS, Version 2.2a, wherever possible.

Table 4-41: Sub-pixel Information Codes (byte offset 0Bh)

Offset	Value	Description/Format
	7 6 5 4 3 2 1 0	Bits
0B_h	00_h	Sub-pixel layout is not defined.
	01_h	Red/Green/Blue vertical stripes (whether or not these stripes are continuous across vertical dimension of display screen).
	02_h	Red/Green/Blue horizontal stripes (whether or not these stripes are continuous across horizontal dimension of display screen).
	03_h	Vertical stripes (as in 01h), with the primary ordering given by the order of the chromaticity information in the DisplayID structure.
	04_h	Horizontal stripes (as in 02h), with the primary ordering given by the order of the chromaticity information in the DisplayID structure.
	05_h	Quad sub-pixels; a 2 x 2 structure with a red sub-pixel at top left, a blue sub-pixel at bottom right, and the remaining two sub-pixels green.
	06_h	As in 05h, but with red at bottom left and blue at top right.
	07_h	Delta (triad) RGB sub-pixels.
	08_h	Mosaic (other)
	09_h	Quad sub-pixels; a 2 x 2 sub-pixel structure including one each of red, green, blue, and one additional color, including white, in any order ¹ .
	0A_h	Five sub-pixels, including RGB sub-pixels aligned as in the case of 01h, with two additional sub-pixels located above or below this group ¹ .
	0B_h	Six sub-pixels; as in case of 0Ah, but with three additional colors in addition to the base set ¹ .
	0C_h	Clairvoyante, Inc. PenTile Matrix™ layout ¹ .
	0D_h - FF_h	Reserved for future use.

Note: In the case of layouts providing more than three sub-pixels, the color coordinates of the additional sub-pixels shall be described as specified in Section 4.9.9.

4.9.6 Horizontal and Vertical Dot or Pixel Pitch

As used in this standard, "dot pitch" or "pixel pitch" means the distance from the geometric center of any given dot or pixel on the physical display screen to the geometric center of the nearest adjacent dot

or pixel along the specified axis (horizontal or vertical). If either of these values is zero, it shall be assumed that there are no discrete dot or pixel structures in the display screen along that axis as, in the case of an aperture-grille CRT in which the grille apertures are aligned with the vertical axis of the tube so the vertical pitch is zero. If both pitch values are given as zero, it shall be assumed that the display device in question does not have a discrete pixel structure (e.g., a monochrome CRT).

In the case of a display which does not have a fixed orientation (e.g., a display which may be pivoted to either a landscape or portrait format device), the meaning of "horizontal" and "vertical" shall be based on the default orientation of the product, as identifiable elsewhere in the DisplayID structure. If a display has a variable dot or pixel pitch, the average value shall be given here.

A special case exists when the display has been identified through other data within the DisplayID structure specifically, the image size information, as a projection type, and therefore has neither a fixed image size, nor a fixed pixel pitch. In this case, the values of these two bytes shall be interpreted as representing the dot pitch values, expressed as the percentage of the image size in that direction, times 100 (which gives a range of possible values of 0.00 to 2.55% of the image size) Again, zero values shall be interpreted as above.

Stating the horizontal and vertical dot or pixel pitches in this manner should not be interpreted as implying that the display screen comprises a rectilinear array of pixels. The configuration and shape of the pixel and sub-pixel structures are given elsewhere in this data block.

Horizontal and Vertical Dot Pitch is defined as in Table 4-42.

Table 4-312: Horizontal and Vertical Pixel Pitch

Byte	Value	Description/Format
0C_h → 0D_h	7 6 5 4 3 2 1 0	Pixel Pitch 0.00 → 2.55 m
0C_h	00_h → FF_h	Horizontal pitch, in increments of 0.01 mm (range 0.00 to 2.55 mm) or 0.01% as above for projection displays
0D_h	00_h → FF_h	Vertical pitch, in increments of 0.01 mm (range 0.00 to 2.55 mm) or 0.01% as above for projection displays

4.9.7 Color Bit Depth

This byte is used to indicate the color "bit depth" (i.e., effective dynamic range) provided for each color. It is assumed that the bit depth of all primaries (typically at least red, green, and blue) are the same. Bits 3-0 give the bit depth, minus 1 (range 1-16) per primary which is provided per frame without considering the effects of frame-to-frame modulation or "temporal dithering" at the display device (e.g., LCD panel or similar).

Table 4-43: Color Bit Depth

Byte	Value	Description/Format
0E_h	7 6 5 4 3 2 1 0	Color Bit Depth
	0_h → F_h	RESERVED set to 0
	— — — — 0_h → F_h	Color Bit Depth – Display Device Values 0 → 15

4.9.8 Response Time

This byte is used to indicate the response time of the display device associated with this DisplayID structure. The value to be stored here shall be the worst-case response time for the display in question,

for any possible transition between two output luminance or reflectance states per measurement method of VESA FPDM 2.

Table 4-324: Response Time

Byte	Value								Description/Format	
0F _h	7	6	5	4	3	2	1	0		
	0	-	-	-	-	-	-	-	Bit 7. If this bit is cleared ('0'), then the value given here is for a "black to white" (lower gray value to higher) transition; if set ('1'), the value is for a "white to black" (higher to lower) transition.	
	-	00_h → 7F_h								Bit 6-0. Response time value (ms). If the response time is less than 1ms, a value of "0" shall be stored here. Similarly, if the response time is any value greater than 126ms, a value of "127" shall be stored here.

4.9.9 Sub-pixel Layout Examples

The Display Device Data Block provides a means for conveying the sub-pixel layout (i.e., the arrangement of individual color areas within a physical pixel) of the display device in question as defined in section 4.9.5. If there is any conflict between information provided here and contents of Section 4.9.5, then Section 4.9.5 shall take precedence.

Code 01_h applies to either individual pixels (left) or continuous vertical stripes (right).

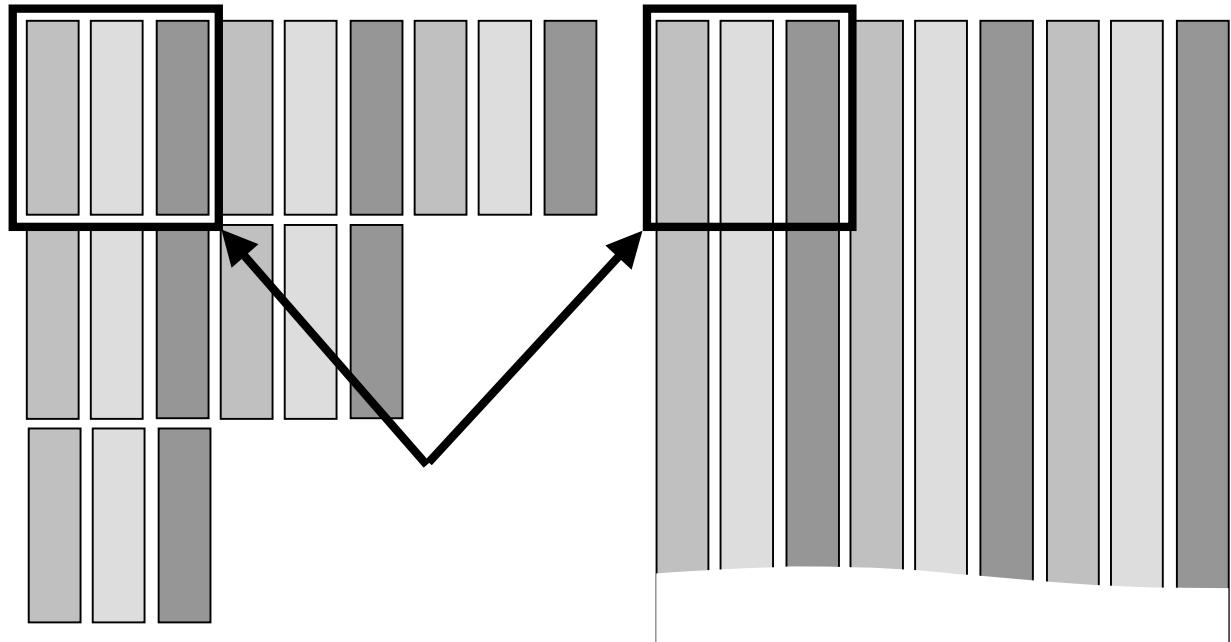


Figure 4-2: Example Code 01_h – RGB Vertical Stripes

Code 02_h applies to either individual pixels (left) or continuous horizontal stripes (right).

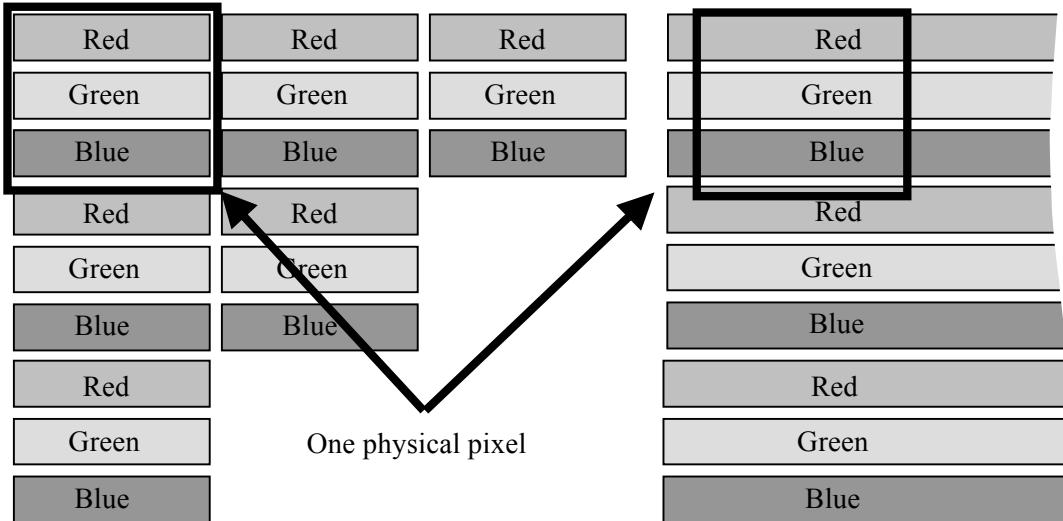


Figure 4-3: Example Code 02_h – RGB Horizontal Stripes

These codes refer to pixel layouts which are physically identical to those described as (01_h) and (02_h), above. The ordering of the primaries are not as shown. Instead, ordering will be as determined by the order of chromaticity information provided in the DisplayID structure, as shown below.

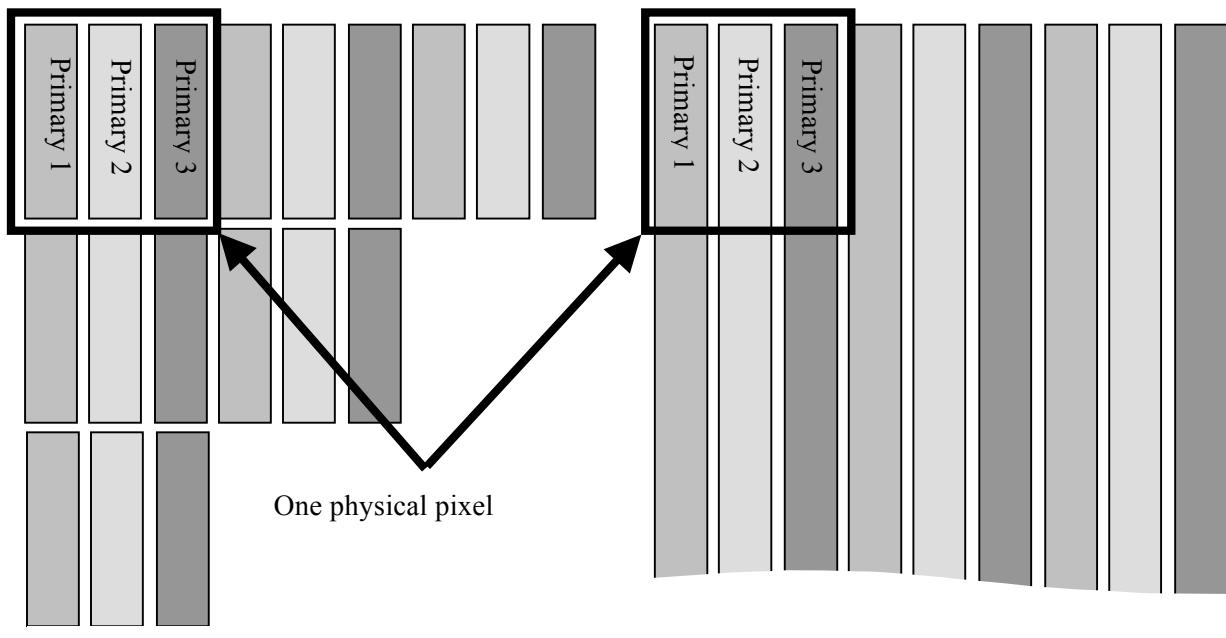


Figure 4-4: Example Codes 03h, 04h– Vertical/Horizontal Stripes with Non-standard Primary Ordering

4.9.9.2

These codes refer to pixel layouts which use a 2×2 square array of sub-pixels. They differ in relative location of the primaries, as shown below.

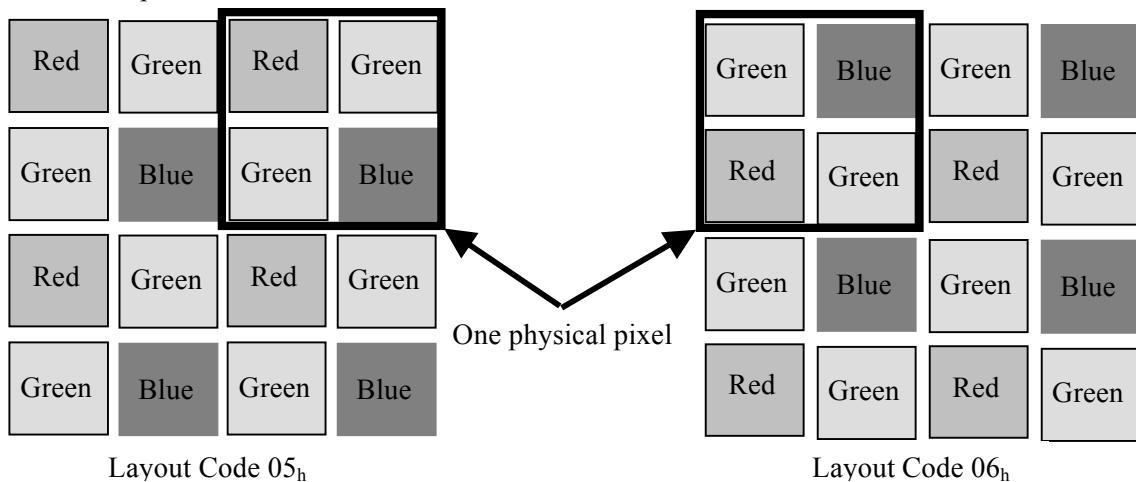


Figure 4-5: Example Codes 05_h, 06_h – Quad Sub-pixels

4.9.9.3

This code refers to layouts involving tri-color (RGB) sub-pixels, either round or square, arranged in a “delta” pattern as shown below.

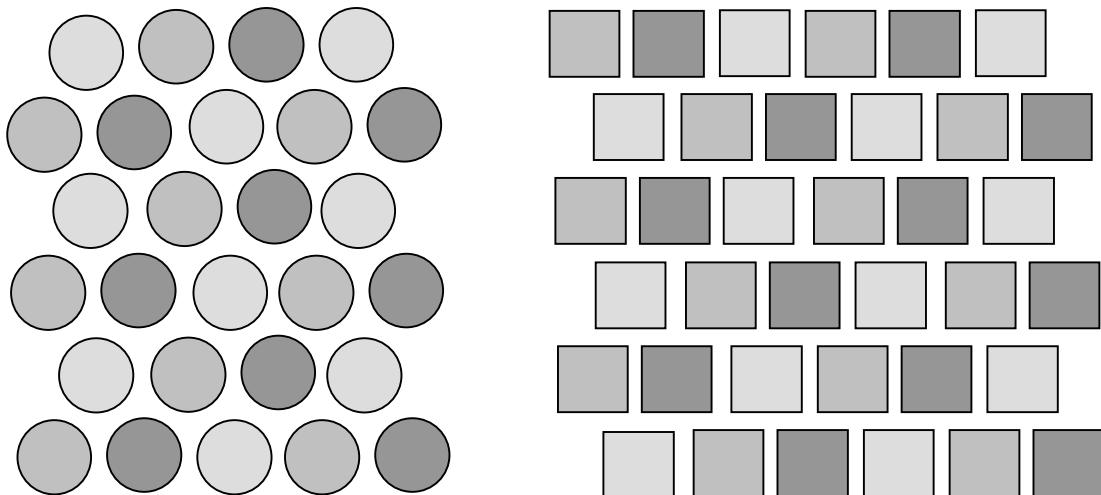


Figure 4-6: Example Code 07_h – Delta Sub-pixels

4.9.9.4 Code 08h – Mosaic (other)

This code is used to identify other sub-pixel layouts which do not conform to any of the standard descriptions given here.

Code 09_h – Quad Sub-pixels

This code describes a 2 x 2 sub-pixel structure including one each of the typical R, G, and B primaries, in the order given by the order of chromaticity (1, 2, 3) as specified in the DisplayID structure, plus one additional color, which may be of any other color, including white, with the chromaticity given in the “Additional Primary Chromaticity” section of this block, as “Primary 4”).

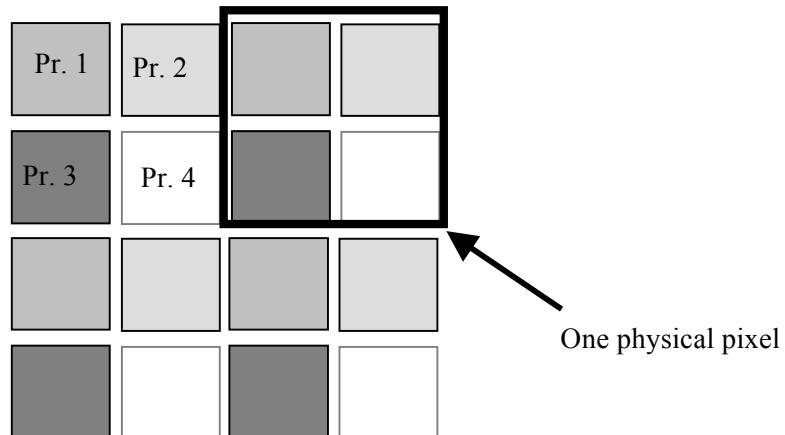


Figure 4-7: Example Code 09_h – Quad Sub-pixels, RGBW or RGBx

4.9.9.5

This code describes a physical pixel which is comprised of five sub-pixels.

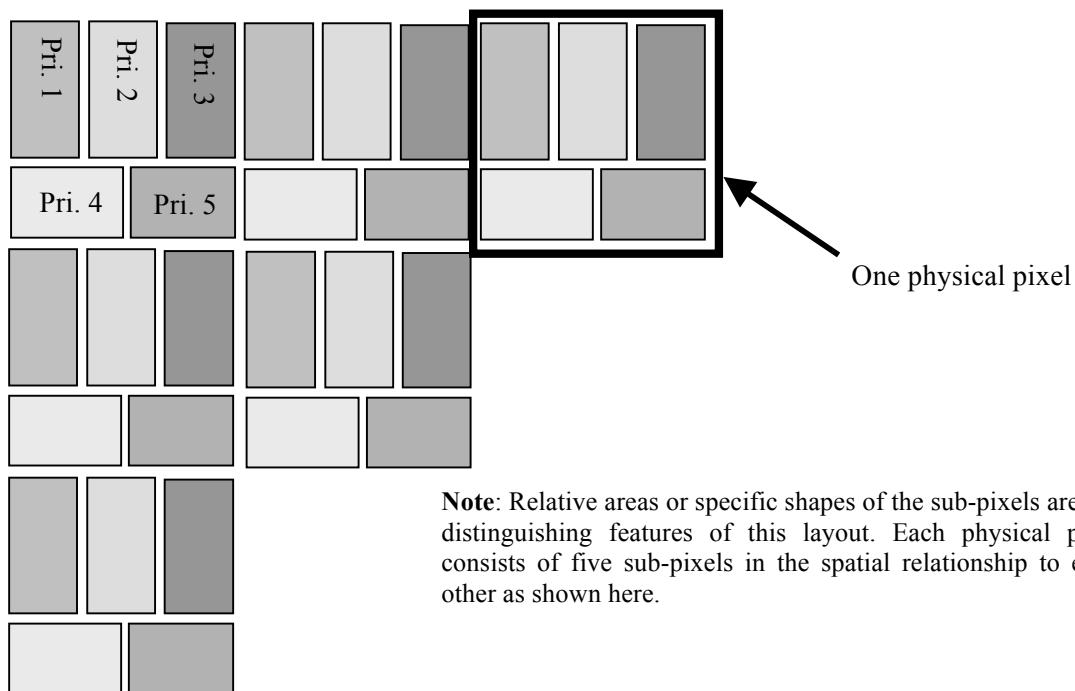
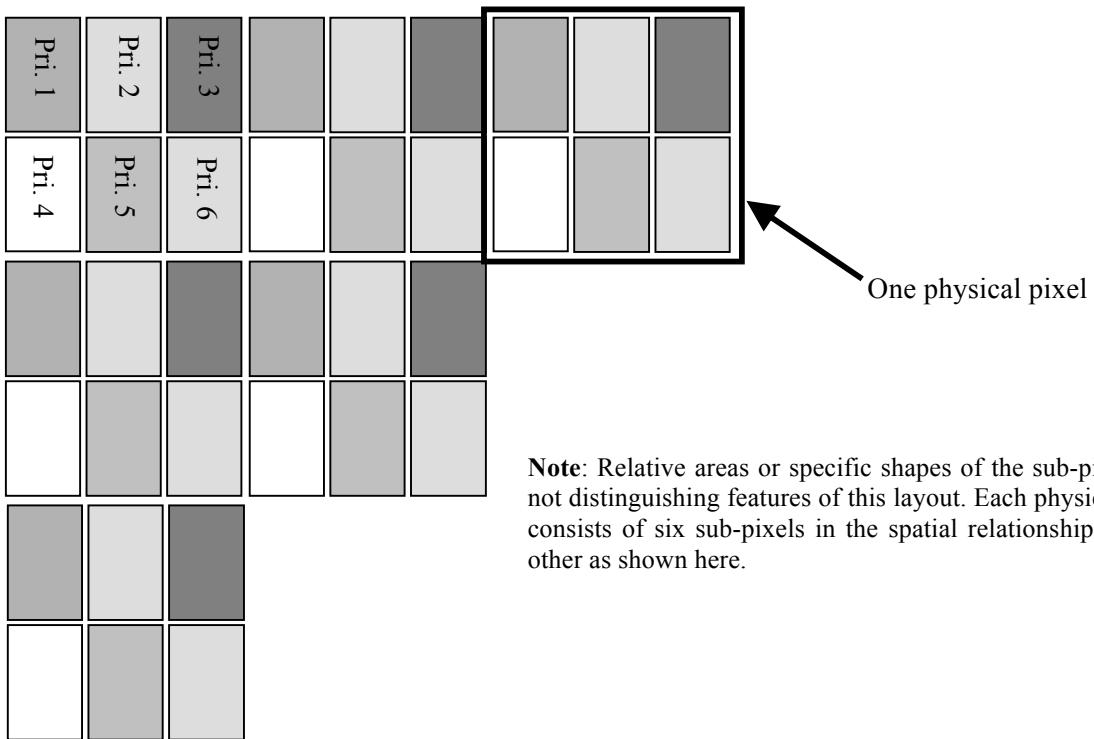


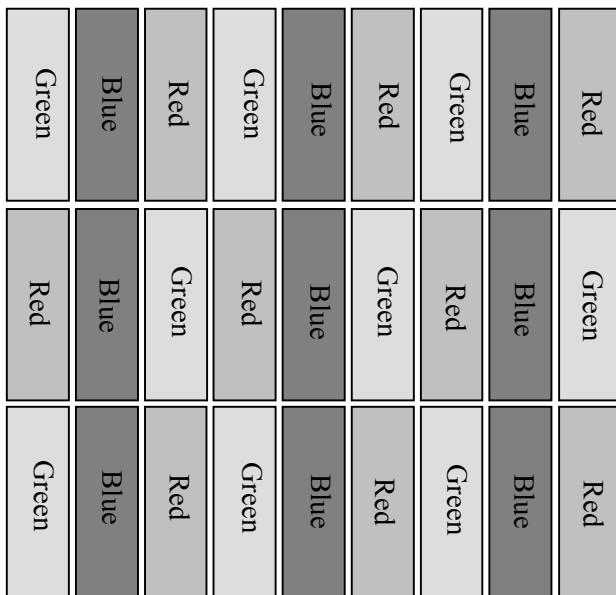
Figure 4-8: Example Code 0A_h – Five Sub-pixels, 3 + 2 Layout



Note: Relative areas or specific shapes of the sub-pixels are not distinguishing features of this layout. Each physical pixel consists of six sub-pixels in the spatial relationship to each other as shown here.

Figure 4-9: Example Code 0B_h – Six Sub-pixels

4.9.9.6



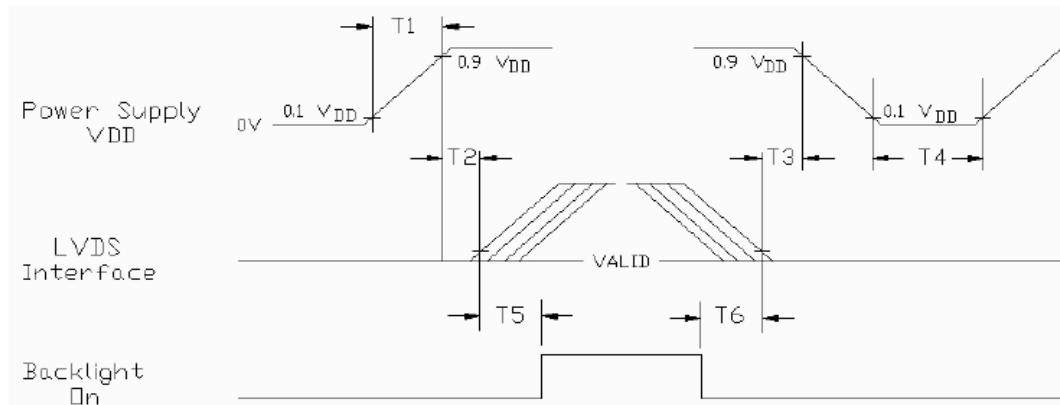
Note: In the Clairvoyante system, there is no *fixed* assignment of a given set of sub-pixels to form a unique physical pixel; sub-pixels are used to create complete “physical” pixels in various combinations, under the control of the drive electronics.

Figure 4-10: Example Code 0C_h – Clairvoyante, Inc., PenTile Matrix™

4.10 Interface Power Sequencing Data Block

Tag Code	0D _h		Revision	0		
Block Size	Fixed					
Type Identifier	1	2	3	4	5	6
Required (R)	O	O	O	O	O	O
Optional (O)	Only one Interface Power Sequencing block is allowed in any DisplayID structure.					

For power sequence delays T1-T6 definitions, please refer to figure below (provided as an example). Interface Power Sequencing data block is to be defined for products that use similar power sequencing scheme.



$0.5 \text{ ms} \leq T_1 \leq 10 \text{ ms}$ $0 \leq t_2 \leq 50 \text{ ms}$ $0 \leq t_3 \leq 50 \text{ ms}$ $500 \text{ ms} \leq T_4 \leq 200 \text{ ms}$ $200 \text{ ms} \leq T_5 \leq 200 \text{ ms} \leq T_6$

Figure 4-11: Interface Power Up/Down Sequence

Table 4-33 shows the detailed description of the block.

Table 4-335: Interface Power Sequencing Data Block Format

Offset	Value								Description/Format/Priority
00 _h	0D _h								Interface Power Sequencing
01 _h	7 6 5 4 3 2 1 0								BLOCK Revision and other Notes
	0 0 0 0								REVISION '0' $0 \rightarrow 2$
	0 0 0 0								RESERVED BITS $3 \rightarrow 7$
02 _h	06 _h								Number of Payload Bytes in BLOCK 6 All Other Values RESERVED
03 _h	7 6 5 4 3 2 1 0								Power Sequence T1 Range Information
	0 _h → F _h - - - -								Power Sequence T1 min = (Bits[7→4] / 10) milliseconds T1 min range is from 0 to 1.5ms with an increment of 0.1ms.
	0 _h → F _h - - - -								Power Sequence T1 max = (Bits[3→0] * 2) milliseconds T1 max range is 0 to 30ms with an increment of 2ms.
04 _h	7 6 5 4 3 2 1 0								Power Sequence T2 Range Information
	0 0 - - - - - -								Bits[7→6] : RESERVED (Set to 0)
	- - - - - -								Power Sequence T2 min is assumed to be 0; Power Sequence T2 max = (Bits[5→0] * 2) milliseconds T2 range is from 0 to 126ms with an increment of 2ms.

05_h	7	6	5	4	3	2	1	0	T3 Range Information
	0	0	—	—	—	—	—	—	Bits[7→6] : RESERVED (Set to 0)
	—	—	00_h → 3F_h				Power Sequence T3 min is assumed to be 0; Power Sequence T3 max = (Bits[5→0] * 2) milliseconds T3 range is from 0 to 126ms with an increment of 2ms.		
06_h	7	6	5	4	3	2	1	0	Power Sequence T4 Min Information
	0	—	—	—	—	—	—	—	Bits[7] : RESERVED (Set to 0)
	—	00_h → 7F_h				Power Sequence T4 min = (Bits[6→0] * 10) milliseconds T4 min range is from 0 to 1270ms with an increment of 10ms.			
07_h	7	6	5	4	3	2	1	0	Power Sequence T5 Min Information
	0	0	—	—	—	—	—	—	Bits[7→6] : RESERVED (Set to 0)
	—	—	00_h → 3F_h				Power Sequence T5 min = (Bits[5→0] * 10) milliseconds T5 range min is from 0 to 63ms with an increment of 10ms.		
08_h	7	6	5	4	3	2	1	0	Power Sequence T6 Min Information
	0	0	—	—	—	—	—	—	Bits[7→6] : RESERVED (Set to 0)
	—	—	00_h → 3F_h				Power Sequence T6 min = (Bits[5→0] * 10) milliseconds T6 range min is from 0 to 630ms with an increment of 10ms.		

4.11 Transfer Characteristics Data Block

Tag Code	0E _h			Revision	1		
Block Size	Variable						
Type Identifier	1		2		3		4
Required (R) Optional (O)	O		O		O		O
Restrictions	More than one Transfer Characteristics Data Block may appear in any given DisplayID structure but each block must describe a unique response curve for a white point or a color primary. No DisplayID structure is presently required to provide a Transfer Characteristics Data Block. Its use is completely optional.						

The Transfer Characteristics Data Block is an optional block that can be used in any DisplayID structure to provide a more detailed description of the display's transfer characteristic ("gamma curve," etc.) than would otherwise be possible. Use of this block is strongly recommended for any device whose response does not follow the traditional "gamma" sort of power function, and therefore cannot be adequately described via a simpler model.

The Transfer Characteristics Data Block provides information describing the transfer characteristic, in terms of output luminance vs. input signal level. This information is given in the form of a base value plus a series of incremental values, all normalized such that the maximum (white level) value is given as 3FF_h (i.e., all luminance points provided are known to 10-bit accuracy, relative to the white level).

The definition of this block follows, as closely as possible, that used in the VESA DTCDB Standard, which was created for use in EDID Extensions. The most significant difference between that definition and this one for DisplayID, is that the number of sample points per curve is not restricted, and the number of primaries whose response curves may be individually described has been increased to six in addition to an overall "white" response. As in the earlier standard, it is required that all transfer characteristic curves described by this block be monotonically increasing; i.e., each successive sample point within the curve must be equal to or greater in luminance than the one preceding it.

Table 4-346: Transfer Characteristics Data Block

Offset	Value								Description/Format/Priority																																			
00_h	0E_h								Transfer Characteristics																																			
01_h	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>0</td><td>0</td><td>1</td> </tr> <tr> <td colspan="4" style="text-align: center;">0_h - F_h</td><td>—</td><td>—</td><td>—</td><td>—</td> </tr> <tr> <td colspan="8" style="text-align: center;">0_h</td> </tr> </table>		7	6	5	4	3	2	1	0	—	—	—	—	—	0	0	1	0 _h - F _h				—	—	—	—	0 _h								BLOCK Revision and other Notes									
7	6	5	4	3	2	1	0																																					
—	—	—	—	—	0	0	1																																					
0 _h - F _h				—	—	—	—																																					
0 _h																																												
									REVISION '1' 0 → 2																																			
									Transfer Characteristics Identifier 0 _h if this block is not associated with a particular color characteristic block. 1 _h - F _h to associate this block with a particular color characteristic block. A corresponding Color Characteristic block is mandatory if this value is non-zero.																																			
02_h	01_h → F8_h								Number of Payload Bytes in BLOCK 1 → 248																																			
	All Other Values RESERVED																																											
03_h	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>1</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td> </tr> <tr> <td>—</td><td>1</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td> </tr> <tr> <td colspan="8" style="text-align: center;">1</td> </tr> <tr> <td colspan="8" style="text-align: center;">0 0 0 0 0 0</td> </tr> </table>		7	6	5	4	3	2	1	0	1	—	—	—	—	—	—	—	—	1	—	—	—	—	—	—	1								0 0 0 0 0 0								Transfer Characteristics Information	
7	6	5	4	3	2	1	0																																					
1	—	—	—	—	—	—	—																																					
—	1	—	—	—	—	—	—																																					
1																																												
0 0 0 0 0 0																																												
									Bit 7: If set, the first curve in this data block (starting at offset 04h) will be the "white" or overall transfer characteristic for the display described by this DisplayID structure (i.e., the response curve observed when all inputs are driven simultaneously and identically from the "black" level to the "white" level).																																			
									Bit 6: If set, the data block provides individual response curves for each input or primary, in the order of primaries listed elsewhere in the DisplayID structure, or in the order of inputs as appropriate for the interface standard in use.																																			
									Bit 5: If set, the data block uses the "four-parameter" model consisting of four eight-bit values plus a gamma exponent value. The number of samples for each of the following response curves must be exactly 5.																																			
									Bits 5→3: RESERVED set to 0.																																			
04_h	N								Number of samples for 1 st response curve for Piecewise-Linear data format. For "four-parameter" data format this value must be set to 5.																																			
05_h	Initial luminance value, 1 st curve or A ₀								Data format depends on Bit 5 in Byte 03 _h . See text below for a description of how the values are stored for each curve.																																			
06_h	First increment (1 st curve) or A ₁																																											
07_h	Second increment (1 st curve) or A ₂																																											
...	...																																											
N+4_h	Final Nth increment (1 st curve)																																											
N+4+1_h	M								Number of samples for 2 nd response curve for Piecewise-Linear data format. For "four-parameter" data format this value must be set to 5.																																			
N+4+2_h	Initial luminance value, 2 nd curve or A ₀																																											
N+4+3_h	First increment (2 nd curve) or A ₁								Data format depends on Bit 5 in Byte 03 _h . See text below for a description of how the values are stored for each curve.																																			
N+4+4_h	Second increment (2 nd curve) or A ₂																																											
...	...																																											
N+4+1 M_h	Final Mth increment (2 nd curve)																																											

Payload bytes: As this block can contain varying numbers of response curves, each of which may have any number of sample points, this is a variable-length block. The value of the payload length byte may be determined as:

Payload bytes = 1 + (Number of “white” curve samples, if provided) + (Number of primary curves provided) * (Number of samples per primary curve).

For "four-parameter" data format, the number of samples is fixed at 5 and includes A₀, A₁, A₂, A₃ and Gamma exponent values in that order.

Flag bits: (Bits 7 and 6 of byte 02_h): Either bit 7 or bit 6, or both, must be set in any valid Transfer Characteristics Data Block.

(Bit 5 of byte 02_h): If not set, then the piecewise-linear data format is selected for the data format. If set, then the four-parameter data format is selected.

Piecewise-Linear Data Format: Each transfer characteristic curve described in this block is given in the form of an initial 8-bit normalized luminance value (for sample “0,” assumed to be the luminance at the “black” input level) followed by a specified number of 8-bit incremental values. Each successive value is added to the total to obtain the luminance level for that sample point, up to a maximum value of 3FFh – which is the normalized value for the final sample, or “white level” luminance, in all cases.

Example of Usage

To better understand how this method is to be used to convey transfer characteristic information, the following examples may be helpful. The values given here are only examples; the actual contents of the data block for any given display will vary depending on the size and type of transfer characteristics chosen for use by the manufacturer, and the specific characteristics of the display in question.

For the purpose of this example, it is assumed that a Display Transfer Characteristic Data Block is to be constructed which provides only a white display transfer characteristic. This is to be stored in the form of sixteen white luminance points, taken for input video signal values spread evenly between the minimum allowable (i.e., the defined "black level" for the video interface in question) and maximum allowable ("white level") values. This applies to either a standard VGA analog interface or an 8-bit/primary digital interface. The luminance data which has been (hypothetically) measured for this display is as follows in Table 4-47.

Table 4-357: Example Luminance Measurements

Sample No.	Analog input level (all inputs)	Digital input value (all inputs)	White Luminance (cd./m ²)
0 (black)	0.000V	00000000	0.56
1	0.047	00010001	3.21
2	0.093	00100010	7.45
3	0.140	00110011	12.62
4	0.187	01000100	18.79
5	0.233	01010101	24.63
6	0.280	01100110	33.98
7	0.327	01110111	40.55
8	0.373	10001000	47.92
9	0.420	10011001	58.41
10	0.467	10101010	65.10
11	0.513	10111011	77.84
12	0.560	11001100	85.33
13	0.607	11011101	91.06
14	0.653	11101110	97.51
15 (peak white)	0.700V	11111111	100.00

To convert these values for storage, the peak white value must first be normalized to 3FFh:

$$100.00 \text{ cd/m}^2 = 3FF_h, \text{ therefore } 1 \text{ lsb (least significant bit)} = 100.00/1023 = 0.098 \text{ cd/m}^2$$

This fixes both the maximum and minimum values; the peak white level value is by definition 3FF_h, and therefore does not need to be conveyed by this data block. The black level luminance (which would be considered the “offset” in the typical gamma + offset response model) is therefore, for this display:

$0.560/0.098 = 5.714$ (normalized), which will be rounded to 6 and stored in first sample location as 00000110. (Assuming this is the first transfer characteristic stored in this data block, this value would be placed in byte 05_h). The remaining values, for samples 2 through 14, are given by storing eight-bit values which correspond to the incremental increase in luminance between samples N and N-1. In other words, the absolute luminance value at any sample point may be determined by summing up all of the eight-bit incremental values up to and including the increment for that sample, and multiplying the resulting value by the lsb luminance. In storing these increments, it shall be the norm to convert the absolute luminance value for any given sample to its ten-bit normalized equivalent first, rounding as necessary, and then determining the incremental value between it and the normalized absolute luminance of the sample immediately preceding. This is shown in Table 4-47**Error! Reference source not found.**, again based on the previous table of luminance samples:

Table 4-48: Example Values for DTCDB Block White Curve, Resulting Measurements as Given Above

Sample #	Luminance	Normalized Luminance	Value to be Stored
0	0.56	0000000110 (006_h)	00000110 (06_h)
1	3.21	0000100001 (021_h)	00011011 ($1B_h$)
2	7.45	0001001100 ($04D_h$)	00101100 ($2C_h$)
3	12.62	0010000001 (081_h)	00110100 (34_h)
4	18.79	0011000000 ($0C0_h$)	00111111 ($3F_h$)
5	24.63	0011111100 ($0FC_h$)	00111100 ($3C_h$)
6	33.98	0101011100 ($15C_h$)	01100000 (60_h)
7	40.55	0110011111 ($19F_h$)	01000011 (43_h)
8	47.92	0111101010 ($1EA_h$)	01001011 ($4B_h$)
9	58.41	1001010110 (256_h)	01101100 ($6C_h$)
10	65.10	1010011010 ($29A_h$)	01000100 (44_h)
11	77.84	1100011100 ($31C_h$)	10000010 (82_h)
12	85.33	1101101001 (369_h)	01001101 ($4D_h$)
13	91.06	1110100100 ($3A4_h$)	00101011 ($3B_h$)
14	97.51	1111100110 ($3E6_h$)	01000010 (42_h)
15	100.00	1111111111 ($3FF_h$)	Not stored

Therefore, for this example, the complete Display Transfer Characteristic Data Block would be as follows:

Table 4-49: Complete DTCDB Contents Using Example Values Given Above

Byte (offset)	Value	Meaning
00 _h	0E _h	Display Transfer Characteristic Data Block tag code
01 _h	00 _h	Rev. number (bits 2-0) is 000. Bits 7-5 are zero indicating this data block is not associated with a specific color characteristic data block.
02 _h	12 _h	Payload length (17 bytes).
03 _h	80 _h	Bit 7 is set to 1, as this block provides a white transfer characteristic. Bit 6 is set to 0, as no primary characteristics are provided. Bits 5-0 are reserved at 0.
04 _h	10 _h	Number of samples, white transfer characteristic curve. (16)
05 _h	06 _h	Initial (black-level) luminance value, normalized as above.
06 _h	1B _h	Increments for samples 2-14. The increment for sample 15 is not needed, as sample 15 is assumed to have a normalized luminance value of 3FFh (the “white” level).
07 _h	2C _h	
08 _h	34 _h	
09 _h	3F _h	
0A _h	3C _h	
0B _h	60 _h	

0C _h	43 _h
0D _h	4B _h
0E _h	6C _h
0F _h	44 _h
10 _h	82 _h
11 _h	4D _h
12 _h	3B _h
13 _h	42 _h

Four-Parameter Gamma Response Curve

In this model, tone response or “gamma” information is provided in the form of four eight-bit values, plus a “gamma” or exponent value, as follows. Refer to the figure provided below.

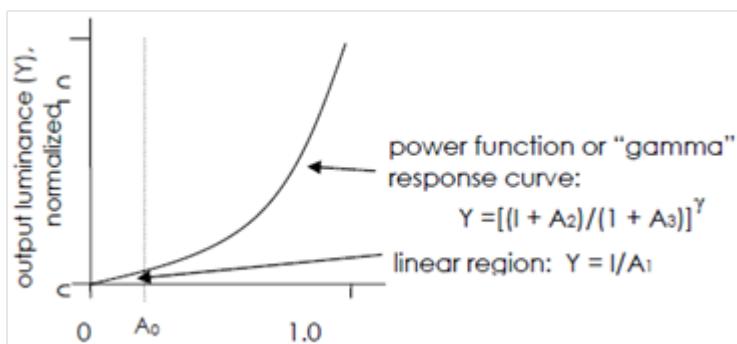


Figure 4-12: Four-Parameter Response Curve

A generalized model for the display response curve which would accommodate addition of a linear section as shown requires the specification of four parameters in addition to the “gamma” exponent value:

For input values (I) $< A_0$: $Y = I/A_1$

For input values (I) $> A_0$: $Y = [(I + A_2)/(1 + A_3)]^\gamma$

The data in the payload consists of five bytes which list the values of A_0 , A_1 , A_2 , A_3 and Gamma exponent (γ) in that order such as the example in Table 4-50.

Table 4-360: Example DTCDB Contents Using "Four-Parameter" Format

Byte (offset)	Value	Meaning
00 _h	0E _h	Display Transfer Characteristic Data Block tag code
01 _h	00 _h	Rev. number (bits 2-0) is 000. Bits 7-5 are zero indicating this data block is not associated with a specific color characteristic data block.
02 _h	07 _h	Payload length (17 bytes).
03 _h	A0 _h	Bit 7 is set to 1, as this block provides a white transfer characteristic. Bit 6 is set to 0, as no primary characteristics are provided. Bit 5 indicates data is in "four-parameter" format. Bits 4-0 are reserved at 0.
04 _h	05 _h	Number of samples, white transfer characteristic curve. (5)
05 _h	00 _h → FF _h	A_0

06 _h	00 _h → FF _h	A ₁
07 _h	00 _h → FF _h	A ₂
08 _h	00 _h → FF _h	A ₃
09 _h	78 _h	Gamma exponent stored as ($\gamma - 1$) * 100 allowing values in the range of 1.00 to 3.55.

4.12 Display Interface Data Block

Tag Code	0F _h		Revision	0		
Block Size	Fixed					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	R	R	R	O	R
Restrictions	No more than one Display Interface Data Block may be provided in any DisplayID structure.					

The Display Interface Data Block is a required data block for product type 2, 3, 4 and 6. It provides information specific to the physical interface between source and sink and is defined in Table 4-51.

Table 4-51: Display Interface Data Block

Offset	Value	Description/Format	
00 _h	0F _h	DISPLAY INTERFACE DATA BLOCK	TAG
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — — 0 0 0	REVISION ‘0’	VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED	
02 _h	0A _h	Number of Payload Bytes in BLOCK	10
03 _h	_DESCRIPTOR	Interface Type/Number of Links or Channels	
04 _h	_DESCRIPTOR	Interface version & revision number	
05 _h	_DESCRIPTOR	Supported Interface color depth for RGB encoding	
06 _h	_DESCRIPTOR	Supported Interface color depth for YCbCr 444 encoding	
07 _h	_DESCRIPTOR	Supported Interface color depth for YCbCr 422 encoding	
08 _h	_DESCRIPTOR	Supported Content Protection	
09 _h	_DESCRIPTOR	Content protection version & revision number	
0A _h	_DESCRIPTOR	Spread Spectrum Information	
0B _h	_DESCRIPTOR	Interface Type dependent Attribute (1)	
0C _h	_DESCRIPTOR	Interface Type dependent Attribute (2)	

Note: In the case there is discrepancy between information in this block and information contained within the interface configuration data (e.g. DPCD for DisplayPort), the source shall use interface configuration data.

4.12.1 Interface Type/Number of Links or Channels

Interface Type: The first (upper) four bits of the byte defined in Table 4-52 shall contain a code, per Table 4-53, which identifies the type of interface used by this display on the physical port from which the DisplayID structure containing this block was obtained.

Table 4-52: Interface Type/Number of Links Format

03 _h	7	6	5	4	3	2	1	0	INTERFACE TYPE/NUMBER OF LINKS
	0 _h → F _h								Interface Type
		0 _h :1 _h :							Number of Links 2 _h :4 _h

Table 4-373: Interface Type Codes

03 _h	7	6	5	4	3	2	1	0	INTERFACE TYPE/NUMBER OF LINKS
		0 _h							Analog (see exception below)
		1 _h							LVDS (generic)
		2 _h							TMDS (generic)
		3 _h							RSDS (generic)
		4 _h							DVI-D
		5 _h							DVI-I, analog section
		6 _h							DVI-I, digital section
		7 _h							HDMI-A
		8 _h							HDMI-B
		9 _h							MDDI
		A _h							DisplayPort
		B _h							Proprietary Digital Interface
		C _h → F _h							RESERVED

Number of Links/Channels: The lower four bits of this byte shall contain the number of links or channels provided by the interface identified in the Interface Type bits, with the definition of link or channel per the standard specification of that interface. For example, a DVI interface may provide either one or two channels, as defined by the DVI specification, and therefore the permissible values for this field in the case of a DVI interface is either “1” or “2,” even though each “channel” as defined by the DVI specification comprises three physical data pairs. **Note** that the interface type field also permits definition of a generic TMDS interface, in which case the number of links or channels would be the actual number of physical data pairs used.

An exception to the above is made in the case of the Interface Type bits identifying an analog video interface (code 0h). In this case, the Number of Links bits are redefined to be a subtype code, which identifies the specific analog interface in use per Table 4-54.

Table 4-384: Analog Interface Sub-type Codes

	7	6	5	4	3	2	1	0	INTERFACE TYPE/NUMBER OF LINKS
03 _h		0 _h			0 _h → 2 _h				Analog Interface Sub-type Codes
		0 _h			0 _h				15HD/VGA (VESA EDDC Standard)
					1 _h				VESA NAVI-V (15HD)
					2 _h				VESA NAVI-D
					3 _h → F _h				RESERVED

4.12.2 Interface Standard Version and Release Number

This byte shall contain the version (upper four bits) and release numbers (lower four bits) of the specification or standard document defining the interface associated with the defined interface type as shown in Table 4-55.

Table 4-395: Interface Standard Version/Release

04_h	7	6	5	4	3	2	1	0	INTERFACE STANDARD VERSION/RELEASE	
	0	h	→	F	h				Interface Standard Version	
					0	h	→	F	h	Interface Standard Revision

4.12.3 Supported Interface Color Depth

The Supported Interface Color Depth is defined by three bytes, one for each of RGB, YCbCr 444 and YCbCr 422 encodings. Each byte defines the supported color depths for that particular encoding as shown in Table 4-56.

Table 4-406: Supported Interface Color Depth

05_h	7	6	5	4	3	2	1	0	Supported Interface color depth for RGB encoding
	0	0	—	—	—	—	—	—	Bits[7→6] : RESERVED (Set to 0)
	—	—	0	—	—	—	—	—	Bit[5] : 16 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	0	—	—	—	—	Bit[4] : 14 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	0	—	—	—	Bit[3] : 12 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	0	—	—	Bit[2] : 10 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	—	0	—	Bit[1] : 8 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	—	—	0	Bit[0] : 6 bit per primary color 0 = Not Supported, 1 = Supported
06_h	7	6	5	4	3	2	1	0	Supported Interface color depth for YCbCr 444 encoding
	0	0	—	—	—	—	—	—	Bits[7→6] : RESERVED (Set to 0)
	—	—	0	—	—	—	—	—	Bit[5] : 16 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	0	—	—	—	—	Bit[4] : 14 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	0	—	—	—	Bit[3] : 12 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	0	—	—	Bit[2] : 10 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	—	0	—	Bit[1] : 8 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	—	—	0	Bit[0] : 6 bit per primary color 0 = Not Supported, 1 = Supported
07_h	7	6	5	4	3	2	1	0	Supported Interface color depth for YCbCr 422 encoding
	0	0	0	—	—	—	—	—	Bits[7→5] : RESERVED (Set to 0)
	—	—	—	0	—	—	—	—	Bit[4] : 16 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	0	—	—	—	Bit[3] : 14 bit per primary color 0 = Not Supported, 1 = Supported
	—	—	—	—	—	0	—	—	Bit[2] : 12 bit per primary color

									0 = Not Supported, 1 = Supported
-	-	-	-	-	-	-	<u>0</u>	-	Bit[1] : 10 bit per primary color 0 = Not Supported, 1 = Supported
-	-	-	-	-	-	-	<u>0</u>	-	Bit[0] : 8 bit per primary color 0 = Not Supported, 1 = Supported

4.12.4 Supported Content Protection

Supported Content Protection shall be a single byte which describes the content protection method supported by the monitor as shown in Table 4-57.

Table 4-417: Content Protection

08_h	7	6	5	4	3	2	1	0	Supported Content Protection
0	0	0	0	0	0	_	_	_	Bit[7→3] : RESERVED (Set to 0)
-	-	-	-	-	-	$00_{\text{h}} \rightarrow 03_{\text{h}}$			Bits [2→0] : Content Protection
						0	0	0	0 = No Content Protection Support
						0	0	1	1 = HDCP
						0	1	0	2 = DTCP
						0	1	1	3 = DPCP
-	-	-	-	-	-	$04_{\text{h}} \rightarrow 07_{\text{h}}$			4 → 7 = RESERVED (Should not be used)

4.12.5 Content Protection Standard Version and Release Number

This byte, as shown in Table 4-58, shall contain the version (upper four bits) and release numbers (lower four bits) of the specification or standard document defining the Content Protection Scheme from Table 4-57. If content protection is not supported, version and revision shall be set to 0.

Table 4-58: Content Protection Standard Version/Release

09_h	7	6	5	4	3	2	1	0	CONTENT PROTECTION STANDARD VERSION/RELEASE
	$0_{\text{h}} \rightarrow F_{\text{h}}$								Content Protection Standard Version
		$0_{\text{h}} \rightarrow F_{\text{h}}$							Content Protection Standard Revision

4.12.6 Spread Spectrum Information

This byte contains spread spectrum related information. For no spread support the spread percentage field must be set to 0. For down or center spread, the spread percentage must indicate proper non-zero percentage. Table 4-59 outlines the format for this byte.

Table 4-59: Spread Spectrum Information

0A_h	7	6	5	4	3	2	1	0	Spread Spectrum Information
									Bit[7→6] : Type of Spread Supported 00 = No Spread Support 01 = Down Spread 10 = Center Spread 11 = Reserved
0	0	-	-	-	-	-	-	-	Bits[5→4] : RESERVED (Set to 0)
-	-	0	0	-	-	-	-	-	Bits[3→0] : Spread Percentage Spread Spectrum Percentage = (Bits[3→0]/10) % Range is from 0 to 1.5%.

4.12.7 Interface Type Dependent Attribute

Byte 0B_h and Byte 0C_h of this block are defined based on the interface type indicated in byte 3. Unless specified, the byte shall be used as reserved and set to 0.

For LVDS (generic) byte 0B_h is defined as follows:

Table 4-60: LVDS (generic), Voltage & Color Mapping

0B_h	7	6	5	4	3	2	1	0	LVDS (generic) : Voltage Support & Color Mapping
0	0	0	-	-	-	-	-	-	Bits[7→5] : RESERVED (Set to 0)
-	-	-	0	-	-	-	-	-	Bit[4]: Color Mapping 0 = NS (Normal) mode, 1 = 6 bit compatible mode
-	-	-	-	0	-	-	-	-	Bit[3] : 2.8 V Support 0 = Not Supported, 1 = Supported
-	-	-	-	-	0	-	-	-	Bit[2] : 12 V Support 0 = Not Supported, 1 = Supported
-	-	-	-	-	-	0	-	-	Bit[1] : 5 V Support 0 = Not Supported, 1 = Supported
-	-	-	-	-	-	-	-	0	Bit[0] : 3.3 V Support 0 = Not Supported, 1 = Supported

Bit[4]: Color Mapping

0 = NS (Normal) mode

1 = 6 bit compatible mode

Bit[4]: Color Mapping describes the Data color mapping applied to the additional 4th (for 8-bit panels) and 5th (for 10bit Panels) LVDS data pair.

If the value is 0, then Most Significant Bits (MSBs) are mapped on the 4th (and 5th for 10-bit panels) data pair.

If the value is 1, then Least Significant Bits (lsbs) are mapped on the 4th (and 5th for 10-bit panels) data pair.

For LVDS (generic) byte 0C_h is defined as follows in Table 4-61.

Table 4-61: LVDS (generic), Timing Signal Settings

0C_h	7	6	5	4	3	2	1	0	LVDS (generic): Timing Signal Settings
	0	0	0	0	0	—	—	—	Bits[7→3] : RESERVED (Set to 0)
	—	—	—	—	—	—	0	—	Bit[2] : DE (Data Enable) Mode 0 = DE Mode, 1 = Fixed Mode
	—	—	—	—	—	—	—	0	Bit [1] : DE Polarity based on DE Mode If Bit[2] is = 0 ‘DE Mode’ 0 = Active High, 1 = Active Low
	—	—	—	—	—	—	—	0	If Bit[2] is = 1 ‘Fixed Mode’ 0 = High Signal Level, 1 = Low Signal Level
	—	—	—	—	—	—	—	0	Bit[0] : Shift Clock Data Strobe 0 = Falling Edge, 1 = Rising Edge

For a Proprietary Digital Interface, byte 0B_h is defined as follows in Table 4-62.

Table 4-422: Proprietary Digital Interface, Timing Signal Settings

0B_h	7	6	5	4	3	2	1	0	Proprietary Digital Interface: Timing Signal Settings
	0	0	0	0	0	—	—	—	Bits[7→3] : RESERVED (Set to 0)
	—	—	—	—	—	—	0	—	Bit [2] : DE (Data Enable) Mode 0 = DE Mode 1 = Fixed Mode
	—	—	—	—	—	—	—	0	Bit[1] : DE Polarity based on DE Mode If Bit[2] is = 0 ‘DE Mode’ 0 = Active High, 1 = Active Low
	—	—	—	—	—	—	—	0	If Bit[2] is = 1 ‘Fixed Mode’ 0 = High Signal Level, 1 = Low Signal Level
	—	—	—	—	—	—	—	0	Bit[0] : Shift Clock Data Strobe 0 = Falling Edge, 1 = Rising Edge

Further description of the DE Mode, DE Polarity and Shift Clock Data Strobe is provided below:

‘DE Mode’ Bit [2]: This Bit is used to distinguish between a panel that is controlled by the DE Signal and one that is controlled in Fixed Mode by VSync and HSync. For Fixed Mode, the Level of the DE Signal is then either fixed Low or fixed High.

‘DE Polarity’ Bit [1]: In Fixed Mode this bit specifies if the DE Signal should be set to Low or High. In DE Mode it specifies the Polarity of the DE Signal.

‘Shift Clock Data Strobe’ Bit [0]: Defines signal edge at which the data will be latched.

4.13 Stereo Display Interface Data Block

Tag Code	10 _h		Revision	1		
Block Size	Variable					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	O	O	O	O	O
Restrictions	Multiple Stereo Display Interface Data Blocks with revision 1 may be exposed in a single DisplayID section.					

This data block type is optional, but must be present if the 3D Stereo Support Flags in any Detailed Timing Descriptor indicate that this display is capable of displaying stereoscopic images. It provides information on the data format required to send stereo image pairs across the interface to the display. Table 4-63 defines the format for this data block.

Table 4-433: Stereo Display Interface Data Block

Offset	Value								Description/Format
00 _h	10 _h								STEREO DISPLAY INTERFACE DATA BLOCK TAG
01 _h	7	6	5	4	3	2	1	0	BLOCK Revision and Other Data
	-	-	-	-	0	0	0	1	REVISION '1' VALUES 0 → 15
	7	6	-	-	-	-	-	-	3D Stereo Timing Support
	0	0	-	-	-	-	-	-	Timings that explicitly reports 3D capability
	0	1	-	-	-	-	-	-	Timings that explicitly reports 3D capability & Timing Code listed as part of the Stereo Display Interface Data Block
	1	0	-	-	-	-	-	-	All listed timing in any of timing block
	1	1	-	-	-	-	-	-	Only Timing Code listed as part of the Stereo Display Interface Data Block
	-	-	5	4	3	-	-	-	RESERVED
	-	-	0	0	0	-	-	-	RESERVED
02 _h	02 _h → FF _h								Number of Payload Bytes in BLOCK N+2
03 _h	01 _h → FF _h								Number of bytes in stereo interface method N+1
04 _h	_DESCRIPTOR								Stereo Interface Method Code Section 4.13.1
05 _h	_DESCRIPTOR								Interface Method Specific Parameters (e.g. N bytes)
	_DESCRIPTOR								..
N+04 _h	_DESCRIPTOR								
N+05 _h	7	6	5	4	3	2	1	0	3D Timing Descriptor only exists if bit 6 of 3D Stereo Timing Support is = '1'
	7	6	-	-	-	-	-	-	Timing Code Type VALUES 0 → 3
	0	0	-	-	-	-	-	-	DMT Timing Code
	0	1	-	-	-	-	-	-	CEA VIC Timing Code
	1	0	-	-	-	-	-	-	HDMI VIC Timing Code
	1	1	-	-	-	-	-	-	RESERVED Timing Code Type
	-	-	5	-	-	-	-	-	RESERVED
	-	-	0	-	-	-	-	-	
	-	-	-	4	3	2	1	0	Number of supported Timing Code

The Stereo Display Interface Data Block revision 0 contains one sub-block that describes the nature of the interface format to the stereoscopic display. The byte at offset 03_h specifies the length of this sub-block, so that other sub-blocks may be added in future. The sub-block contains a Stereo Interface Method Code (1 byte) and a set of Interface Method specific parameters. Stereo Display Interface Data Block revision 1 adds provision for optionally exposing supported timing descriptor for a given 3D stereo interface descriptor. 3D Stereo Timing Support field is used to indicate to what timing a given stereo method is applied.

The table below describes the bit values and the corresponding timing support. Whenever the field indicates timing code listed as part of the Stereo Display Interface Data Block, the block will have additional timing descriptor following the Interface Method Specific Parameters field.

The 3D Stereo Timing Support field default of ‘0’ indicates that this Data Block only applies to those timing descriptors which explicitly report 3D capability. Possible values for the 3D Stereo Timing Support field can be found in the table below.

Table 4-444: 3D Stereo Timing Support

Value		Description	
7	6	3D Stereo Timing Support	
0	0	Timings that explicitly reports 3D capability	
0	1	Timings that explicitly reports 3D capability & Timing Code listed as part of the Stereo Display Interface Data Block	
1	0	All listed timing in any of timing block	
1	1	Only Timing Code listed as part of the Stereo Display Interface Data Block	

Refer to Appendix E for an example of how the data block can be used to expose different types of stereo method and the associated timings.

4.13.1 Stereo Interface Method Code

The Stereo Interface Method Code field is a single byte that encodes the nature of the interface, as defined in Table 4-65. Depending on the Code value, there are 1 to 8 bytes of additional parameters describing further details on the interface data format. The parameter definitions are found in the sections indicated.

Table 4-455: Stereo Interface Method Codes

Code	Interface Method	Parameter Bytes	Section
00 _h	Field Sequential Stereo	1	Section 4.13.2
01 _h	Side-by-side Stereo	1	Section 4.13.3
02 _h	Pixel Interleaved Stereo	8	Section 4.13.4
03 _h	Dual Interface, Left and Right Separate	1	Section 4.13.5
04 _h	Multi-view	2	Section 4.13.6
05 _h	Stacked Frame Stereo	1	Section 4.13.7
05 _h : FE _h	RESERVED		
FF _h	Proprietary	0	Section 4.13.8

4.13.2 Field Sequential Stereo

This Stereo Interface Method indicates that the display expects an image sequence consisting of Left, Right, Left, etc. The display includes a selection device (shutter glasses, Z-screen, etc.) which derives left/right information through a VESA Standard Connector for Stereoscopic Display Hardware or through another standard method. (Other proprietary interfaces will be supported for legacy devices only).

There is one parameter byte associated with the Field Sequential Stereo interface method. The lsb of this parameter byte indicates the left/right Stereo Sync signal polarity. A ‘0’ in the polarity bit indicates that Stereo Sync is logical ‘1’ on transmission of a ‘left-eye’ image, and ‘0’ on a ‘right-eye’ image. A ‘1’ indicates that Stereo Sync is ‘1’ on transmission of a ‘right-eye’ image, and ‘0’ on a ‘left eye’ image.

For a device using a VESA Standard Connector for Stereoscopic Display Hardware, the polarity is ‘0’. For other methods the recommended polarity is ‘0’, i.e. the Stereo Sync signal is ‘1’ on transmission of a left image.

Table 4-466: Stereo Display Interface Data Block for Field Sequential Stereo

Offset	Value	Description/Format
00 _h	10 _h	STEREO DISPLAY INTERFACE DATA BLOCK TAG
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data
	— — — — 0 0 0 1	REVISION ‘1’ VALUES 0 → 15
	0 — — — — — — —	3D Stereo Timing Support 0: Only timings that explicitly reports 3D capability 1: All timing descriptors
	— 0 0 0 — — — —	RESERVED
02 _h	03 _h	Number of Payload Bytes in BLOCK 3
03 _h	02 _h	Number of bytes in stereo interface method section 2

04_h	00_h	Field Sequential Stereo
05_h	_DESCRIPTOR	Stereo Polarity (lsb)

4.13.3 Side-by-side Stereo

This Stereo Interface Method indicates that the display expects a single image, the left half of which corresponds to a first of the stereo pair and the right half to the second of the stereo pair. Table 4-67 defines the format for this interface method.

There is no scaling assume; e.g. an 800x600 display that takes side-by-side data would declare a detailed timing descriptor of resolution 1600x600 marked as ‘always displayed in stereo’.

There is a single parameter byte associated with the Side-by-side Stereo Interface Method. The lsb of the parameter indicates the identity of the views. If ‘0’, the left half of the image represents the ‘left eye’ view and the right half represents the ‘right eye’ view. If ‘1’, the left half of the image represents the ‘right eye’ view and the right half the ‘left eye’ view.

Table 4-477: Stereo Display Interface Data Block for Side-by-side Stereo

Offset	Value								Description/Format																																	
00_h	10_h								STEREO DISPLAY INTERFACE DATA BLOCK	TAG																																
01_h	<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>—</td><td>—</td><td>—</td><td>—</td><td>0</td><td>0</td><td>0</td><td>1</td> </tr> <tr> <td>0</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td> </tr> <tr> <td>—</td><td>0</td><td>0</td><td>0</td><td>—</td><td>—</td><td>—</td><td>—</td> </tr> </table>								7	6	5	4	3	2	1	0	—	—	—	—	0	0	0	1	0	—	—	—	—	—	—	—	—	0	0	0	—	—	—	—	BLOCK Revision and Other Data	
7	6	5	4	3	2	1	0																																			
—	—	—	—	0	0	0	1																																			
0	—	—	—	—	—	—	—																																			
—	0	0	0	—	—	—	—																																			
									REVISION ‘1’	VALUES 0 → 15																																
									3D Stereo Timing Support																																	
									0: Only timings that explicitly reports 3D capability 1: All timing descriptors																																	
									RESERVED																																	
02_h	03_h								Number of Payload Bytes in BLOCK	3																																
03_h	02_h								Number of bytes in stereo interface method section	2																																
04_h	01_h								Side-by-side Interleaved Stereo																																	
05_h	_DESCRIPTOR								View Identity (lsb)	0 or 1																																

4.13.4 Pixel Interleaved Stereo

This Stereo Interface Method indicates that the display expects a single image, in which pixels from the left-eye view and right-eye view are interleaved in a repeating pattern. The repeating pattern is defined by the eight parameter bytes. Table 4-68 defines the format for this interface method.

The eight parameter bytes define an 8x8 pixel pattern that starts at the leftmost, topmost pixel of the display and is horizontally and vertically repeated until the right side respectively bottom of the display is reached.

Table 4-68: Stereo Display Interface Data Block for Pixel Interleaved Stereo

Offset	Value								Description/Format	
00 _h	10 _h								STEREO DISPLAY INTERFACE DATA BLOCK	TAG
01 _h	7 6 5 4 3 2 1 0								BLOCK Revision and Other Data	
	— — — — 0 0 0 1								REVISION ‘1’	VALUES 0 → 15
	0 — — — — — — —								3D Stereo Timing Support	
	0: Only timings that explicitly reports 3D capability 1: All timing descriptors									
	— 0 0 0 — — — —								RESERVED	
02 _h	0A _h								Number of Payload Bytes in BLOCK	10
03 _h	09 _h								Number of bytes in stereo interface method section	9
04 _h	02 _h								Pixel Interleaved Stereo	
05 _h : 0C _h	DESCRIPTOR								8x8 Interleave Pattern (see description below)	

Each bit in the Interleave Pattern descriptor, when ‘1’ indicates that the pixel position is a ‘left-eye image pixel’, when ‘0’ indicates a ‘right-eye image pixel’.

The first byte of the interleave pattern describes the topmost line, the next byte the second line etc. Bit 7 of each byte describes the leftmost image pixel. Bit 6 describes the first pixel from the left, etc.

Note: The EDID 1.3 standard defines stereo viewer types ‘2-way interleaved stereo, right image on even lines’ and ‘2-way interleaved stereo, left image on even lines’. These types can be represented easily in the Pixel Interleaved Stereo Interface Method. For example, ‘2-way interleaved stereo, right image on even lines’ can be represented by parameter values: FF_h, 00_h, FF_h, 00_h, FF_h, 00_h, FF_h, 00_h.

4.13.5 Dual Interface Left and Right Separate

This Stereo Interface Method applies to display devices that have dual interfaces. One interface carries the image sequence for the left-eye view. The other carries the right-eye view. In order to support a wide variety of stereo display techniques, this interface method supports optional mirroring. The 1-byte parameter associated with this interface method specifies whether the interface to which the DisplayID structure applies carries the left or right view and specifies the optional mirroring to be applied.

Table 4-69: Stereo Display Interface Data Block for Dual Interface Left and Right Separate

Offset	Value								Description/Format	
00 _h	10 _h								STEREO DISPLAY INTERFACE DATA BLOCK	TAG
01 _h	7 6 5 4 3 2 1 0								BLOCK Revision and Other Data	
	— — — — 0 0 0 1								REVISION ‘1’	VALUES 0 → 15
	0 — — — — — — —								3D Stereo Timing Support	
	0: Only timings that explicitly reports 3D capability 1: All timing descriptors									
	— 0 0 0 — — — —								RESERVED	
02 _h	03 _h								Number of Payload Bytes in BLOCK	3
03 _h	02 _h								Number of bytes in stereo interface method section	2
04 _h	03 _h								Dual Interface Left and Right Separate	
05 _h	DESCRIPTOR								Polarity and Mirroring (see description)	

The lsb (bit 0) of the parameter byte indicates Left/Right polarity:

- ‘1’ – this interface carries the Left-eye view
- ‘0’ – this interface carries the Right-eye view

Bits 2 & 1 indicate the desired mirroring, as follows:

- ‘00’ – no mirroring
- ‘01’ – left/right mirrored
- ‘10’ – top/bottom mirrored
- ‘11’ – RESERVED

Note: If the source device encounters two ‘left’ or two ‘right’ view devices, it will arbitrarily assign one the ‘left’ and the other the ‘right’ view.

4.13.6 Multi-view

This indicates a display with more than two views, e.g. a lenticular or barrier type auto-stereoscopic display. There are two parameter bytes associated with this stereo display interface type. The first parameter is the number of discrete full color views that are required to compose the image to drive this display. The second parameter byte is a code for the view interleaving method.

The view interleaving method code is assigned by VESA at the request of a display manufacturer. The manufacturer must provide VESA with a description of the view interleaving method at the time of the request.

Table 4-70: Stereo Display Interface Data Block for Multi-view

Offset	Value								Description/Format	
00 _h	10 _h								STEREO DISPLAY INTERFACE DATA BLOCK TAG	
01 _h	7 6 5 4 3 2 1 0								BLOCK Revision and Other Data	
	— — — — 0 0 0 1								REVISION ‘1’ VALUES 0 → 15	
	0 — — — — — — —								3D Stereo Timing Support 0: Only timings that explicitly reports 3D capability 1: All timing descriptors	
	— 0 0 0 — — —								RESERVED	
02 _h	04 _h								Number of Payload Bytes in BLOCK 4	
03 _h	03 _h								Number of bytes in stereo interface method section 3	
04 _h	04 _h								Multi-view	
05 _h	DESCRIPTOR								Number of views > 2	
06 _h	DESCRIPTOR								View Interleaving Method Code VALUES 0 → 255	

4.13.7 Stacked Frame Stereo

This Stereo Interface Method indicates that the display expects a single image, the top portion of which corresponds to the first of the stereo pair and the bottom portion to the second of the stereo pair. Between the two image portions are additional unused image lines whose count is equal to the vertical blank of the timing.

There is no scaling assumed; e.g. an 800x600 display with a vertical blank of 36 lines that takes stacked frame stereo would declare a detailed timing descriptor of resolution 800x600. This detailed timing will be used as-is for monoscopic timings. For stereo timings the resolution output to the display

should be increased to 800x1236; 800x600 for top and bottom portions plus an additional 36 unused lines between them.

There is a single parameter byte associated with the Stacked Frame Stereo Interface Method. The first bit (lsb) of the parameter byte indicates the identity of the views. The value must be set to ‘0’ to indicate the top portion of the image represents the ‘left eye’ and bottom portion represents the ‘right eye’. Other values are reserved and must not be used.

Table 4-71: Stereo Display Interface Data Block for Stacked Frame Stereo

Offset	Value	Description/Format	
00 _h	10 _h	STEREO DISPLAY INTERFACE DATA BLOCK	TAG
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — 0 0 0 1	REVISION ‘1’	VALUES 0 → 15
	0 — — — — — — —	3D Stereo Timing Support	
	0 — — — — — — —	0: Only timings that explicitly reports 3D capability 1: All timing descriptors	
	— 0 0 0 — — — —	RESERVED	
02 _h	03 _h	Number of Payload Bytes in BLOCK	3
03 _h	02 _h	Number of bytes in stereo interface method section	2
04 _h	05 _h	Stacked Frame Stereo	
05 _h	DESCRIPTOR	View Identity (LSB)	0

4.13.8 Proprietary Stereo Interface Methods

For stereo displays that use a proprietary interface method, a Stereo Display Interface Data Block is required. The Stereo Interface Method Code should be set to FF_h. The vendor should include elsewhere in the DisplayID structure a Vendor-Specific Data Block that contains the stereo parameters needed for the proprietary display.

Table 4-482: Stereo Display Interface Data Block for Proprietary Stereo Interface Methods

Offset	Value	Description / Format	
00 _h	10 _h	STEREO DISPLAY INTERFACE DATA BLOCK	TAG
01 _h	7 6 5 4 3 2 1 0	BLOCK Revision and Other Data	
	— — — — 0 0 0 1	REVISION ‘1’	VALUES 0 → 15
	0 — — — — — — —	3D Stereo Timing Support	
	0 — — — — — — —	0: Only timings that explicitly reports 3D capability 1: All timing descriptors	
	— 0 0 0 — — — —	RESERVED	
02 _h	02 _h	Number of Payload Bytes in BLOCK	2
03 _h	01 _h	Number of bytes in stereo interface method section	1
04 _h	FF _h	Proprietary Stereo Interface Method	

4.14 Tiled Display Topology Data Block

Tag Code	12_h		Revision	0		
Block Size	Fixed					
Type Identifier	1	2	3	4	5	6
Required (R) Optional (O)	O	O	O	O	O	O
Restrictions	No restriction is imposed on this block and the payload, apart from the requirement that it conform to format listed below. Current framework allows for multiple Tiled Display Topology Data Block to be defined in a DisplayID block however for revision 0 of this block it is expected that only single Tiled Display Topology Block will be exposed by the sink. In the event that a source device that supports Tiled Display Data Block encounters more than one Tiled Display Topology Data Block with the revision field set to 0h, it must interpret the first such block, and must ignore any subsequent blocks.					

The Tiled Display Data Block is used to describe tiled displays whether they are implemented in a single enclosure or implemented as a collection of separate physical displays configured in a tiled display topology. The source may use the information provided by the Tiled Display Data Block to determine the best way to automatically configure the display and/or to provide useful information to the user about the display configuration.

Full definition of the format of Tiled Display Topology Data Block Structure and Descriptors are given in the following table:

Table 4-73: Tiled Display Topology Data Block

Offset	Value	Description / Format / Priority
00_h	12_h	TILED DISPLAY TOPOLOGY DATA BLOCK TAG
01_h	7 6 5 4 3 2 1 0	TILED DISPLAY TOPOLOGY DATA BLOCK REVISION AND OTHER DATA
	— — — — — 0 0 0	REVISION '0' VALUES 0 → 7
	0 0 0 0 0 — — —	RESERVED
02_h	16_h , ...	Tiled Display Topology Data Block Number of Payload Bytes in BLOCK = 22
03_h	_DESCRIPTOR	Tiled Display and Tile Capabilities Section 4.14.14.1
04_h, 06_h	_DESCRIPTOR	Tiled Display Topology Section 4.1
05_h, 06_h	_DESCRIPTOR	Tile Location Section 4.14.34.1
07_h → 0A_h	_DESCRIPTOR	Tile Size Section 4.14.44.1

0B_h → 0F_h	_DESCRIPTOR	Tile Pixel Multiplier & Tile Bezel Information Section 4.14.3
10_h → 18_h	_DESCRIPTOR	Tiled Display Topology ID Section 4.14.6 4.1

4.14.1 Tiled Display and Tile Capabilities

Table 4-74: Tiled Display and Tile Capabilities Description

03_h	7	6	5	4	3	2	1	0	Tiled Display and Tile Capabilities								
	1	—	—	—	—	—	—	—	0= The tiled display consists of multiple physical display enclosures 1= The tiled display is within a single physical display enclosure.								
	—	—	1	—	—	—	—	—	0 = Tile Bezel Information descriptor is not available, 0B _h - 0E _h shall be set to zero. Note: Pixel Multiplier must be zero for this case. 1 = Offset 0B _h - 0E _h contains Tile Bezel Information descriptor. This also requires Pixel Multiplier field at offset 0A _h to be non-zero For a tile display consisting of multiple physical display enclosures, this bit must be 1.								
			5														
	—	—	0	—	—	—	—	—	Reserved								
	—	—	—	4	3	2	1	0	Behavior of this tile when subsets of the tiles of the entire tiled display are receiving images from source. Note: If all tiles are receiving images from source, each image is displayed at the location specified by tile location field.								
				4	3				Behavior when more than 1 tile and less than total number of tiles are driven by the source Note: When the total number of tiles is 2, then this field (Bits 4:3) becomes “don’t care” as the number of tiles driven by the source cannot be more than 1 and less than the total number.								
					0 _h				0= Behavior cannot be described by other values defined in this revision of the data block 1= When this tile is receiving an image, the image is displayed at the location specified by the Tile Location 2-3 = Reserved								
						2	1	0	Behavior of this tile when it is the only tile receiving an image from source								
	—	—	—	—	—	—			0= Behavior cannot be described by other values defined in this revision of the data block 1= Image is displayed at the location specified by the Tile Location 2= Image is scaled to fit entire tiled display 3= Image is cloned to all other tiles of the entire tiled display 4-7= Reserved								
	—	—	—	—	—	—	0 _h →7 _h										

Note: If a sink exposes a non-native mode in EDID or DisplayID, the source may send that mode as is and sink must handle stitching of the tiles across the seam appropriately for a single enclosure display case. The stitching must work across all supported scaling options exposed by the sink for that resolution in the OSD. If the sink can't handle stitching across the seam with scaled image for a given resolution then it should either not report the mode in EDID or if mode is reported in EDID it should center the image and not expose scaling option in the OSD for that resolution.

4.14.2 Tiled Display Topology

Table 4-75: Tiled Display Topology

04_h,06_h	0_h → 3F_h								Total Number of Horizontal Tiles 1 → 64 Bits [3:0] located at offset 4_h and Bits [5:4] located at offset 6_h.
04_h	7	6	5	4	3	2	1	0	
	0_h → F_h	—	—	—	—	—	—	—	Total Number of Horizontal Tiles (Low Bits [3:0])
04_h,06_h	0_h → 3F_h								Total Number of Vertical Tiles 1 → 64 Bits [3:0] located at offset 4_h and Bits [5:4] located at offset 6_h.
	7	6	5	4	3	2	1	0	
	—	—	—	—	0_h → F_h	—	—	—	Total Number of Vertical Tiles (Low bits [3:0])

06_h	7	6	5	4	3	2	1	0	Tiled Display Topology (high bits) & Tile Location (high bits)
	0_h	—	—	—	—	—	—	—	Total Number of Horizontal Tiles (High bits [5:4])
	→	3 _h	—	—	—	—	—	—	
	—	—	0_h	—	—	—	—	—	Total Number of Vertical Tiles (High bits [5:4])
	—	—	—	—	0_h	—	—	—	Horizontal Tile Location (High bits [5:4])
	—	—	—	—	→	3 _h	—	—	
	—	—	—	—	—	—	0_h	—	Vertical Tile Location (High bits [5:4])
	—	—	—	—	—	—	→	3 _h	

These fields describe the tiled display topology. The lower four bits of the field are located at offset 4 and higher 2 bits are located at offset 6.

4.14.3 Tile Location

Table 4-76: Tile Location

05_h,06_h	0_h → 3F_h								Horizontal Tile Location Bits [3:0] located at offset 5 _h and Bits [5:4] located at offset 6 _h .	1 → 64	
05_h	7	6	5	4	3	2	1	0			
	0 _h → F _h	—	—	—	—	—	—	—	Horizontal Tile Location (Low Bits [3:0])		
05_h,06_h	0_h → 3F_h								Vertical Tile Location Bits [3:0] located at offset 5 _h and Bits [5:4] located at offset 6 _h .	1 → 64	
	7	6	5	4	3	2	1	0			
	—	—	—	—	0 _h → F _h	—	—	—	Vertical Tile Location (Low Bits [3:0])		

06_h	7	6	5	4	3	2	1	0	Tiled Display Topology (high bits) & Tile Location (high bits)		
	0 _h	—	—	—	—	—	—	—	Total Number of Horizontal Tiles (High bits [5:4])		
	→	—	—	—	—	—	—	—			
	3 _h	—	—	—	—	—	—	—			
	—	—	0 _h	—	—	—	—	—	Total Number of Vertical Tiles (High bits [5:4])		
	—	—	→	—	—	—	—	—			
	—	—	3 _h	—	—	—	—	—	Horizontal Tile Location (High bits [5:4])		
	—	—	—	—	0 _h	—	—	—			
	—	—	—	—	→	—	—	—	Vertical Tile Location (High bits [5:4])		
	—	—	—	—	3 _h	—	—	—			

These fields describe the location of the individual tile within the topology. The lower four bits of the field are located at offset 5 and higher 2 bits are located at offset 6.

4.14.4 Tile Size

Table 4-77: Tile Size

Offset	Value	Tile Size	
7_h, 8_h	00 00_h → FF FF_h	Horizontal Size	1 → 65,536 Pixels
7 _h	00 _h → FF _h	Low bits 7 → 0	
8 _h	00 _h → FF _h	High bits 15 → 8	
9_h, A_h	00 00_h → FF FF_h	Vertical Size	1 → 65,536 Lines
9 _h	00 _h → FF _h	Low bits 7 → 0	
A _h	00 _h → FF _h	High bits 15 → 8	

These fields describe the size of a tile (not the size of the entire tiled display) at native resolution.

4.14.5 Tile Pixel Multiplier & Tile Bezel Information

Table 4-78: Tile Pixel Multiplier & Tile Bezel Information

Offset	Value	Pixel Multiplier & Tile Bezel Information	
0B_h	00_h → FF_h	Pixel Multiplier	0 → 255
0C_h	00_h → FF_h	Top Bezel Size	0 → 255
		Top Bezel in pixels = (Pixel Multiplier x Top Bezel Size x 0.1)	
0D_h	00_h → FF_h	Bottom Bezel Size	0 → 255
		Bottom Bezel in pixels = (Pixel Multiplier x Bottom Bezel Size x 0.1)	
0E_h	00_h → FF_h	Right Bezel Size	0 → 255
		Right Bezel in pixels = (Pixel Multiplier x Right Bezel Size x 0.1)	
0F_h	00_h → FF_h	Left Bezel Size	0 → 255
		Left Bezel in pixels = (Pixel Multiplier x Left Bezel Size x 0.1)	

These fields describe pixel multiplier and bezel related information. If bezel information is indicated to be present by setting bit 6 of the Tiled Display and Tile Capability field to “1” then Pixel Multiplier must be non-zero. Bezel size in pixels calculation was originally based on pixel multiplier representing a unit of pixels per cm and bezel Size in mm, however any generic multiplier can be used to get higher pixel precision if required.

4.14.6 Tiled Display Topology ID

Tiled Display Topology ID descriptor comprises of Tiled Display Manufacturer/Vendor ID, Product ID and Serial Number field. These fields will be used by the source to uniquely identify the Topology for both single enclosure and external display topologies. These fields may match the Vendor ID, Product ID and Serial Number fields exposed elsewhere in EDID or DisplayID but is not required to be so. Associating tiles with a specific Tiled Display is required in order to produce the expected behavior when more than one Tiled Display is connected to a source device. Without this correct association, tiles from different Tiled Displays may be grouped together by the source, resulting in tiled images being sent to, and displayed on, the wrong physical display device.

Table 4-79: Tiled Display Topology ID Descriptor

Offset $10_h \rightarrow 18_h$	Value	Tiled Display Topology ID
$10_h \rightarrow 12_h$	_DESCRIPTOR	Tiled Display Vendor ID
$13_h \rightarrow 14_h$	_DESCRIPTOR	Tiled Display Product Code
$15_h \rightarrow 18_h$	_DESCRIPTOR	Tiled Display Serial Number

4.14.6.1 Tiled Display Manufacturer/Vendor ID

Tiled display Manufacturer/Vendor ID field is a required element of the Tiled Display Topology ID descriptor. The Manufacturer/Vendor ID name field, shown below, contains the display manufacturer's three character code. The field contains three bytes identifying the display's manufacturer or vendor. This is the same as the ISA (Industry Standard Architecture) Plug 'N' Play ID for that company.

ISA manufacturer PNPs are issued by Microsoft. Contact Microsoft by e-mail, fax, or website at:

E-mail: pnpid@microsoft.com
 Fax: 425-936-7329, Attention: PNPPID in Building 27
 URL: <http://www.microsoft.com/whdc/system/pnppwr/pnp/pnpid.mspx>

Table 4-80: Tiled Display Manufacturer/Vendor ID Format

Offset	Value	Description
10_h	ASCII Code	Character 1
11_h	ASCII Code	Character 2
12_h	ASCII Code	Character 3

4.14.6.2 Tiled Display Product Code

Tiled display product code is a required element of the Tiled Display Topology ID descriptor. The product code field, shown in Table 4-2, contains a 2-byte vendor-assigned product code. This is used to differentiate between different models from the same manufacturer. If this field is used to represent a Product ID Code (for example a model number), then the number shall be stored in hex format with the least significant byte listed first.

Table 4-81: Tiled Display Product Code

Offset	Value	Description
13 _h	00 _h → FF _h	Product ID Code: LSB
14 _h	00 _h → FF _h	Product ID Code: MSB

4.14.6.3 Tiled Display Serial Number

Tiled display serial number is a required element of the Tiled Display Topology ID descriptor. The serial number is a 32-bit serial number used to differentiate between individual instances of the same model of display product. When used, the bit order for this field follows that what is shown below. The four bytes of the serial number are listed with least significant byte (LSB) first. The range of this serial number is 0 to 4,294,967,295. This serial number is a number only and is not intended to represent ASCII codes.

Table 4-82: Serial Number Format

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
15 _h	7	6	5	4	3	2	1	0	ID Serial Number
16 _h	15	14	13	12	11	10	9	8	
17 _h	23	22	21	20	19	18	17	16	
18 _h	31	30	29	28	27	26	25	24	

4.15 Vendor-Specific Data Block

Tag Code	7F _h		Revision	0			
Block Size	Variable						
Type Identifier	1	2	3	4	5	6	
Required (R) Optional (O)	O	O	O	O	O	O	
Restrictions	No restriction is imposed on this block and the payload apart from the requirement that it conform to format listed below and that it must use a valid Vendor ID.						

The Vendor-specific Data Block is to be used for proprietary implementation not supported under this standard. This block is only intended to be used when the required data cannot be conveyed by a standard data block defined by VESA or CEA. The vendor is then encouraged to bring a proposal for a new standard data block to VESA.

The block must conform to the following format to ensure generic parser of DisplayID structure work properly. Number of Payload bytes for this block must include the three bytes needed for Vendor ID in addition to the size of the vendor-specific data.

Table 4-83: Vendor-Specific Data Block

Offset	Value								Description/Format	
00_h	7F_h								VENDOR-SPECIFIC DATA BLOCK	TAG
01_h	7	6	5	4	3	2	1	0	BLOCK Revision and Other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’	VALUES 0 → 7
	0	0	0	0	0	—	—	—	RESERVED	
02_h	03_h → F8_h								Number of Payload Bytes in BLOCK	3 → 248
03_h → 05_h	_DESCRIPTOR								Vendor ID	Section 4.1.1
07_h ...	Vendor Specific Data									

5 Appendix A: DisplayID Example

5.1 DisplayID Example 1 (Version 1.0)

This example contains a variable length section with second byte indicating the section size.

		Field Name	Offset	Offset	Value	Value
			(Dec)	(Hex)	(Hex)	(Dec)
Section	Section Header	Structure Version/Revision	0	0	10	16
		Section Size	1	1	58	88
		Product Type Identifier	2	2	3	3
		Extension Count	3	3	0	0
	Payloads	Block Tag	4	4	0	0
		Block Revision & Other Data	5	5	0	0
		Number of Payload Bytes	6	6	16	22
		Vendor ID	7	7	41	65
		Vendor ID	8	8	44	68
		Vendor ID	9	9	56	86
		Product Code	10	A	A0	160
		Product Code	11	B	2B	43
		Serial Number	12	C	53	83
		Serial Number	13	D	46	70
		Serial Number	14	E	55	85
		Serial Number	15	F	32	50
		Week of Manufacture / Model Tag	16	10	0A	10
		Year of Manufacture / Model Year	17	11	8	8
Display Parameters Data Block	Payloads	Size of Product ID String	18	12	A	10
		Product ID String (optional)	19	13	53	83
		Product ID String (optional)	20	14	61	97
		Product ID String (optional)	21	15	6D	109
		Product ID String (optional)	22	16	70	112
		Product ID String (optional)	23	17	6C	108
		Product ID String (optional)	24	18	65	101
		Product ID String (optional)	25	19	20	32
		Product ID String (optional)	26	1A	44	68
		Product ID String (optional)	27	1B	49	73
		Product ID String (optional)	28	1C	44	68
Block Header	Block Header	Block Tag	29	1D	1	1
		Block Revision & Other Data	30	1E	0	0
		Number of Payload Bytes	31	1F	C	12
	Payloads	Horizontal Image Size	32	20	7	7
		Horizontal Image Size	33	21	2	2
		Vertical Image Size	34	22	40	64

			Vertical Image Size	35	23	1	1
			Horizontal Pixel Count	36	24	80	128
			Horizontal Pixel Count	37	25	7	7
			Vertical Pixel Count	38	26	B0	176
			Vertical Pixel Count	39	27	4	4
			Feature Support Flags	40	28	18	24
			Transfer Characteristic Gamma	41	29	78	120
			Aspect Ratio	42	2A	3C	60
			Color Bit Depth	43	2B	75	117
			Block Tag	44	2C	2	2
			Block Revision & Other Data	45	2D	0	0
			Number of Payload Bytes	46	2E	0D	13
			Color Characteristics Information	47	2F	B1	177
			Color x or u' value low bits	48	30	3D	61
		Payloads	Color x or u' value high bits/Color y or v' value low bits	49	31	7A	122
			Color y or v' value high bits	50	32	54	84
			Color x or u' value low bits	51	33	CC	204
			Color x or u' value high bits/Color y or v' value low bits	52	34	49	73
			Color y or v' value high bits	53	35	99	153
			Color x or u' value low bits	54	36	66	102
			Color x or u' value high bits/Color y or v' value low bits	55	37	25	37
			Color y or v' value high bits	56	38	F	15
			Color x or u' value low bits	57	39	0	0
			Color x or u' value high bits/Color y or v' value low bits	58	3A	5F	95
			Color y or v' value high bits	59	3B	51	81
			Block Tag	60	3C	3	3
			Block Revision & Other Data	61	3D	0	0
		Payloads	Number of Payload Bytes	62	3E	14	20
			Pixel clock ÷ 10,000	63	3F	27	39
			Pixel clock ÷ 10,000	64	40	3C	60
			Pixel clock ÷ 10,000	65	41	0	0
			Timing Options	66	42	85	133
			Horizontal Active Image Pixels	67	43	7F	127
			Horizontal Active Image Pixels	68	44	7	7
			Horizontal Blank Pixels	69	45	9F	159
			Horizontal Blank Pixels	70	46	0	0
			Horizontal Offset (Front Porch)	71	47	2F	47
			Horizontal Offset (Front Porch)	72	48	80	128
			Horizontal Sync Width	73	49	1F	31

			Horizontal Sync Width	74	4A	0	0
			Vertical Active Image Lines	75	4B	AF	175
			Vertical Active Image Lines	76	4C	4	4
			Vertical Blank Lines	77	4D	22	34
			Vertical Blank Lines	78	4E	0	0
			Vertical Sync Offset (Front Porch)	79	4F	2	2
			Vertical Sync Offset (Front Porch)	80	50	0	0
			Vertical Sync Width	81	51	5	5
			Vertical Sync Width	82	52	0	0
	Power Sequencing Data Block	Block Header	Block Tag	83	53	0D	13
			Block Revision & Other Data	84	54	0	0
			Number of Payload Bytes	85	55	6	6
		Payloads	Power Sequence T1 Range Information	86	56	88	136
			Power Sequence T2 Range Information	87	57	20	32
			T3 Range Information	88	58	20	32
			Power Sequence T4 Min Information	89	59	40	64
			Power Sequence T5 Min Information	90	5A	20	32
			Power Sequence T6 Min Information	91	5B	20	32
		Checksum	Checksum	92	5C	8B	139

5.2 DisplayID Example 2 (Version 1.1)

This example contains a fixed length 127 bytes section size with second byte indicating the section size (excluding header + checksum). Unused bytes are filled at the end with fill data 00h and checksum field situated as last byte of the section.

		Field Name		Offset (Dec)	Offset (Hex)	Value (Hex)	Value (Dec)
Product Identification Data Block	Section Header	Structure Version/Revision	0	0	11	17	
		Section Size	1	1	7A	122	
		Product Type Identifier	2	2	3	3	
		Extension Count	3	3	0	0	
Block Header	Payloads	Block Tag	4	4	0	0	
		Block Revision & Other Data	5	5	0	0	
		Number of Payload Bytes	6	6	16	22	
Payloads		Vendor ID	7	7	41	65	
		Vendor ID	8	8	44	68	
		Vendor ID	9	9	56	86	
		Product Code	10	A	A0	160	

			Product Code	11	B	2B	43
			Serial Number	12	C	53	83
			Serial Number	13	D	46	70
			Serial Number	14	E	55	85
			Serial Number	15	F	32	50
			Week of Manufacture / Model Tag	16	10	0A	10
			Year of Manufacture / Model Year	17	11	8	8
			Size of Product ID String	18	12	A	10
			Product ID String (optional)	19	13	53	83
			Product ID String (optional)	20	14	61	97
			Product ID String (optional)	21	15	6D	109
			Product ID String (optional)	22	16	70	112
			Product ID String (optional)	23	17	6C	108
			Product ID String (optional)	24	18	65	101
			Product ID String (optional)	25	19	20	32
			Product ID String (optional)	26	1A	44	68
			Product ID String (optional)	27	1B	49	73
			Product ID String (optional)	28	1C	44	68
			Block Tag	29	1D	1	1
			Block Revision & Other Data	30	1E	0	0
			Number of Payload Bytes	31	1F	C	12
			Horizontal Image Size	32	20	7	7
			Horizontal Image Size	33	21	2	2
			Vertical Image Size	34	22	40	64
			Vertical Image Size	35	23	1	1
			Horizontal Pixel Count	36	24	80	128
			Horizontal Pixel Count	37	25	7	7
			Vertical Pixel Count	38	26	B0	176
			Vertical Pixel Count	39	27	4	4
			Feature Support Flags	40	28	18	24
			Transfer Characteristic Gamma	41	29	78	120
			Aspect Ratio	42	2A	3C	60
			Color Bit Depth	43	2B	75	117
			Block Tag	44	2C	2	2
			Block Revision & Other Data	45	2D	0	0
			Number of Payload Bytes	46	2E	0D	13
			Color Characteristics Information	47	2F	B1	177
			Color x or u' value low bits	48	30	3D	61
			Color x or u' value high bits/Color y or v' value low bits	49	31	7A	122
			Color y or v' value high bits	50	32	54	84
			Color x or u' value low bits	51	33	CC	204

Type I Timing – Detailed Block	Block Header	Color x or u' value high bits/Color y or v' value low bits	52	34	49	73
		Color y or v' value high bits	53	35	99	153
		Color x or u' value low bits	54	36	66	102
		Color x or u' value high bits/Color y or v' value low bits	55	37	25	37
		Color y or v' value high bits	56	38	F	15
		Color x or u' value low bits	57	39	0	0
		Color x or u' value high bits/Color y or v' value low bits	58	3A	5F	95
		Color y or v' value high bits	59	3B	51	81
	Payloads	Block Tag	60	3C	3	3
		Block Revision & Other Data	61	3D	0	0
		Number of Payload Bytes	62	3E	14	20
		Pixel clock ÷ 10,000	63	3F	27	39
		Pixel clock ÷ 10,000	64	40	3C	60
		Pixel clock ÷ 10,000	65	41	0	0
		Timing Options	66	42	85	133
		Horizontal Active Image Pixels	67	43	7F	127
		Horizontal Active Image Pixels	68	44	7	7
		Horizontal Blank Pixels	69	45	9F	159
		Horizontal Blank Pixels	70	46	0	0
		Horizontal Offset (Front Porch)	71	47	2F	47
		Horizontal Offset (Front Porch)	72	48	80	128
		Horizontal Sync Width	73	49	1F	31
		Horizontal Sync Width	74	4A	0	0
		Vertical Active Image Lines	75	4B	AF	175
		Vertical Active Image Lines	76	4C	4	4
		Vertical Blank Lines	77	4D	22	34
		Vertical Blank Lines	78	4E	0	0
	Power Sequencing Data Block	Vertical Sync Offset (Front Porch)	79	4F	2	2
		Vertical Sync Offset (Front Porch)	80	50	0	0
		Vertical Sync Width	81	51	5	5
		Vertical Sync Width	82	52	0	0
		Block Tag	83	53	0D	13
		Block Revision & Other Data	84	54	0	0
	Payloads	Number of Payload Bytes	85	55	6	6
		Power Sequence T1 Range Information	86	56	88	136
		Power Sequence T2 Range Information	87	57	20	32
		T3 Range Information	88	58	20	32
		Power Sequence T4 Min Information	89	59	40	64

		Power Sequence T5 Min Information	90	5A	20	32
		Power Sequence T6 Min Information	91	5B	20	32
		FILL data	92	5C	0	0
		FILL data	93	5D	0	0
		FILL data	94	5E	0	0
		FILL data	95	5F	0	0
		FILL data	96	60	0	0
		FILL data	97	61	0	0
		FILL data	98	62	0	0
		FILL data	99	63	0	0
		FILL data	100	64	0	0
		FILL data	101	65	0	0
		FILL data	102	66	0	0
		FILL data	103	67	0	0
		FILL data	104	68	0	0
		FILL data	105	69	0	0
		FILL data	106	6A	0	0
		FILL data	107	6B	0	0
		FILL data	108	6C	0	0
		FILL data	109	6D	0	0
		FILL data	110	6E	0	0
		FILL data	111	6F	0	0
		FILL data	112	70	0	0
		FILL data	113	71	0	0
		FILL data	114	72	0	0
		FILL data	115	73	0	0
		FILL data	116	74	0	0
		FILL data	117	75	0	0
		FILL data	118	76	0	0
		FILL data	119	77	0	0
		FILL data	120	78	0	0
		FILL data	121	79	0	0
		FILL data	122	7A	0	0
		FILL data	123	7B	0	0
		FILL data	124	7C	0	0
		FILL data	125	7D	0	0
	Checksum	Checksum	126	7E	68	104

6 Appendix B: DisplayID as an EDID Extension

The main requirements of an EDID extension block are as follows:

- Byte 0 is TAG: DisplayID EDID Extension Block Tag (Tag 70_h would be reserved)
- Byte 1 is version: DisplayID version/revision of the section would be used as version field
- Byte 2-126 is data: DisplayID Section header + DisplayID block data + DisplayID Fill Data + DisplayID Section Checksum + EDID Extension Fill Data.
- Byte 127 Checksum: Checksum of the EDID extension.

When using a DisplayID section of less than 126 byte as an EDID extension, unused bytes after the DisplayID checksum shall be initialized to fill data of (00_h) up to the EDID extension checksum byte.

Only a single DisplayID section is permitted per-EDID extension block.

It is recommended that when an EDID that comprises of CEA extension blocks and DisplayID extension blocks that DisplayID extensions are exposed after all of the CEA extensions.

For example an EDID with base EDID, 2 CEA extensions block and a DisplayID extension block is recommended to be setup as follows:

Block 0 => EDID

Block 1 => 1st CEA Extension

Block 2 => 2nd CEA Extension

Block 3 => DisplayID Extension

When DisplayID is used as an extension, all blocks are optional. DisplayID data blocks that convey new information such as new high resolution timing, new stereo 3D fromat, tiled display topology, transfer characteristics, panel power sequence block and others can be exposed in the DisplayID extension.

Example 1 and 2 show how DisplayID extension to a base DisplayID section would be constructed.

Example 3 and 4 show how DisplayID section to base EDID would be constructed using DisplayID fill data or EDID extension fill data.

Example 1: DisplayID Extension to Base DisplayID – variable length, no fill data						
		Field Name	Offset	Offset	Value	
(Dec)	(Hex)		(Hex)	(Hex)	(Dec)	
		Structure Version/Revision	0	00	10	
		Section Size	1	01	09	
		Product Type Identifier	2	02	06	
		Extension Count	3	03	00	
		Block Tag	4	04	0D	
		Block Revision/Tags	5	05	00	
		Number of Payload Bytes	6	06	06	
		T1	7	07	88	
		T2	8	08	20	
		T3	9	09	20	
		T4	10	0A	40	
		T5	11	0B	20	
		T6	12	0C	32	
		DID Section Checksum	13	0D	86	
					134	

Example 2: DisplayID Extension to Base DisplayID – fixed length of 128 bytes				
	Field Name	Offset (Dec)	Offset (Hex)	Value (Hex)
	Structure Version/Revision	0	00	10 16
	Section Size	1	01	7B 123
	Product Type Identifier	2	02	06 6
	Extension Count	3	03	00 0
	Block Tag	4	04	0D 13
	Block Revision/Tags	5	05	00 0
	Number of Payload Bytes	6	06	06 6
	T1	7	07	88 136
	T2	8	08	20 32
	T3	9	09	20 32
	T4	10	0A	40 64
	T5	11	0B	20 32
	T6	12	0C	20 32
	Fill Data	13..126	0D..7E	00 0
	DID Section Checksum	127	7F	14 20

Example 3: DisplayID Extension to Base EDID – fixed length of 128 bytes, exterior fill (recommended)				
	Field Name	Offset (Dec)	Offset (Hex)	Value (Hex)
	EDID Extension Block Tag	0	00	70 112
	Structure Version/Revision	1	01	10 16
	Section Size	2	02	09 9
	Product Type Identifier	3	03	06 6
	Extension Count	4	04	00 0
	Block Tag	5	05	0D 13
	Block Revision/Tags	6	06	00 0
	Number of Payload Bytes	7	07	06 6
	T1	8	08	88 136
	T2	9	09	20 32
	T3	10	0A	20 32
	T4	11	0B	40 64
	T5	12	0C	20 32
	T6	13	0D	20 32
	DID Section Checksum	14	0E	86 134
	Fill Data	15..126	0F..7E	00 0
	EDID Extension Block Checksum	127	7F	90 144

Example 4: DisplayID Extension to Base EDID – fixed length of 128 bytes, interior fill				
	Field Name	Offset (Dec)	Offset (Hex)	Value (Hex)
	EDID Extension Block Tag	0	00	70 112
	Structure Version/Revision	1	01	10 16
	Section Size	2	02	79 121
	Product Type Identifier	3	03	06 6
	Extension Count	4	04	00 0
	Block Tag	5	05	0D 13
	Block Revision/Tags	6	06	00 0
	Number of Payload Bytes	7	07	06 6

	T1	8	08	88	136
	T2	9	09	20	32
	T3	10	0A	20	32
	T4	11	0B	40	64
	T5	12	0C	20	32
	T6	13	0D	20	32
	Fill Data	14..125	0F..7D	00	0
	DID Section Checksum	126	7E	16	22
	EDID Extension Block Checksum	127	7F	90	144

7 Appendix C: CEA Data Block within a DisplayID Data Block

DisplayID Data Block Format

Offset	Value								Description/Format	
00_h	00_h → FF_h								DATA BLOCK IDENTIFICATION	TAG
01_h	7	6	5	4	3	2	1	0	BLOCK Revision and Other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’	VALUES 0 → 7
	0	0	0	0	0	—	—	—	RESERVED (BLOCK SPECIFIC)	FLAGS / TAG
02_h	00_h → F8_h								Number of Payload Bytes 0 → 248	
03_h	_DESCRIPTOR								1st Data Payload Byte	
.	_DESCRIPTOR								2nd Data Payload Byte ... (if present)	
.	.								.	.

Data Block Identification code 81_h-FF_h was reserved for CEA use in the DisplayID specification. Identification code 81_h will be used for DisplayID block that describes how to map CEA data blocks on to a DisplayID data block. The table below describes how to map CEA block with appropriate block tag code, associated block length and the block payload on the DisplayID CEA Data Block. This will allow leverage of CEA 861 defined speaker allocation, audio descriptor, colorimetry data block and others as is without redefining them in DisplayID.

CEA Data Block

Offset	Value								Description/Format	
00_h	81_h								CEA DATA BLOCK IDENTIFICATION	TAG
01_h	7	6	5	4	3	2	1	0	BLOCK Revision and Other Data	
	—	—	—	—	—	0	0	0	REVISION ‘0’	VALUES 0 → 7
	0	0	0	0	0	—	—	—	RESERVED (BLOCK SPECIFIC)	FLAGS / TAG
02_h	00_h → F8_h								Number of Payload Bytes 0 → 248	
03_h	7	6	5	4	3	2	1	0	CEA Block1 Tag Code and Block1 Length	
	7	6	5	—	—	—	—	—	Tag Code	
	0 _h →	—	—	—	—	—	—	—	Based on latest CEA 861 spec	
	—	—	—	4	3	2	1	0	Block Length	
	—	—	—	0 _h → 1F _h	—	—	—	—	Associated data block length (e.g. L1)	
04_h	_DESCRIPTOR								CEA block1 Descriptor 1	
05_h	_DESCRIPTOR								CEA block1 Descriptor 2	
...									...	
1+L1	_DESCRIPTOR								CEA block1 Descriptor L1	
2+L1	7	6	5	4	3	2	1	0	CEA Block2 Tag Code and Block2 Length	
	7	6	5	—	—	—	—	—	Tag Code	
	0 _h →	—	—	—	—	—	—	—	Based on latest CEA 861 spec	
	—	—	—	4	3	2	1	0	Block Length	
	—	—	—	0 _h → 1F _h	—	—	—	—	Associated data block length (e.g. L2)	
3+L1	_DESCRIPTOR								CEA block2 Descriptor 1	
4+L1	_DESCRIPTOR								CEA block2 Descriptor 2	
...									...	
2+	_DESCRIPTOR								CEA block1 Descriptor L2	

L1+L2		

8 Appendix D: Type IV Timing Data Block Example

When the monitor supports the timings as below:

1) DMT Timings are supported with

1920 x 1200 @ 60 RB	(DMT ID 44 _h)
1600 x 1200 @ 60	(DMT ID 33 _h)
1680 x 1050 @ 60 RB	(DMT ID 39 _h)
1440 x 1050 @ 60 RB	(DMT ID 29 _h)
1366 x 768 @ 60	(DMT ID 51 _h)
1280 x 800 @ 60 RB	(DMT ID 1B _h)

2) CEA timings are supported with

1920 x 1080 @ 60	(CEA VIC 16 =>10 _h)
1920 x 1080 @ 50	(CEA VIC 31 =>1F _h)
1280 x 720 @ 60	(CEA VIC 4 =>04 _h)
1280 x 720 @ 50	(CEA VIC 19 =>13 _h)
1920 x 1080 @ 24	(CEA VIC 32 =>20 _h)
1280 x 720 @ 24	(CEA VIC 60 =>3C _h)

3) HDMI timings are supported with

3840 x 2160 @ 24	(HDMI VIC 3 =>03 _h)
4096 x 2160 @ 24	(HDMI VIC 4 =>04 _h)
3840 x 2160 @ 30	(HDMI VIC 1 =>01 _h)

Offset	Value	Bit	Value	Description	
00h	0x06	[7:0]		Type IV Timing (Short)	
01h	0x01	[7:6]	0b00	Timing Description Type	(00 : DMT Code)
		[5:3]	0b000	RESERVED	
		[2:0]	0b001	Revision '1'	
02h	0x06	[7:0]	6	Number of Payload Bytes in Block	
03h	0x44	[7:0]		1920 x 1200 @ 60 RB	(DMT ID 44 _h)
04h	0x33	[7:0]		1600 x 1200 @ 60	(DMT ID 33 _h)
05h	0x39	[7:0]		1680 x 1050 @ 60 RB	(DMT ID 39 _h)
06h	0x29	[7:0]		1440 x 1050 @ 60 RB	(DMT ID 29 _h)
07h	0x51	[7:0]		1366 x 768 @ 60	(DMT ID 51 _h)
08h	0x1B	[7:0]		1280 x 800 @ 60 RB	(DMT ID 1B _h)

Offset	Value	Bit	Value	Description	
00h	0x06	[7:0]		Type IV Timing (Short)	
01h	0x41	[7:6]	0b01	Timing Description Type	(01 : CEA VIC Code)
		[5:3]	0b000	RESERVED	

		[2:0]	0b001	Revision '1'	
02h	0x06	[7:0]	6	Number of Payload Bytes in Block	
03h	0x10	[7:0]		1920 x 1080 @ 60	(CEA VIC 16 =>10 _h)
04h	0x1F	[7:0]		1920 x 1080 @ 50	(CEA VIC 31 =>1F _h)
05h	0x04	[7:0]		1280 x 720 @ 60	(CEA VIC 4 =>04 _h)
06h	0x13	[7:0]		1280 x 720 @ 50	(CEA VIC 19 =>13 _h)
07h	0x20	[7:0]		1920 x 1080 @ 24	(CEA VIC 32 =>20 _h)
08h	0x3C	[7:0]		1280 x 720 @ 24	(CEA VIC 60 =>3C _h)

Offset	Value	Bit	Value	Description	
00h	0x06	[7:0]		Type IV Timing (Short)	
01h	0x81	[7:6]	0b10	Timing Description Type	(10 : HDMI VIC Code)
		[5:3]	0b000	RESERVED	
		[2:0]	0b001	Revision '1'	
02h	0x03	[7:0]	3	Number of Payload Bytes in Block	
12h	0x03	[7:0]		3840 x 2160 @ 24	(HDMI VIC 3 =>03 _h)
13h	0x04	[7:0]		4096 x 2160 @ 24	(HDMI VIC 4 =>04 _h)
14h	0x01	[7:0]		3840 x 2160 @ 30	(HDMI VIC 1 =>01 _h)

9 Appendix E: Stereo Display Interface Data Block Example

When the monitor supports two formats of stereo display methods which are Stacked Frame and Side-by-Side, then it supports several timings for each stereo method as below.

1) Stacked Frame is supported with

- 1920 x 1200 @ 60 RB (DMT ID 44_h)
- 1920 x 1080 @ 60 (DMT ID 52_h)
- 1600 x 1200 @ 60 (DMT ID 33_h)
- 1280 x 720 @ 60 (DMT ID 55_h)
- 1920 x 1080 @ 24 (CEA VIC 32 =>20_h)
- 1280 x 720 @ 24 (CEA VIC 60 =>3C_h)
- 1280 x 720 @ 50 (CEA VIC 19 =>13_h)

2) Side-by-Side is supported with

- 1920 x 1080 @ 24 (CEA VIC 32 =>20_h)
- 1280 x 720 @ 24 (CEA VIC 60 =>3C_h)
- 1280 x 720 @ 60 (CEA VIC 4 =>04_h)
- 1280 x 720 @ 50 (CEA VIC 19 =>13_h)
- 1920 x 1080 @ 60i (CEA VIC 5 =>02_h)
- 1920 x 1080 @ 50i (CEA VIC 20 =>14_h)

Offset	Value	Bit	Value	Description	
00h	0x10	[7:0]		Stereo Display Interface Data Block	
01h	0xC1	[7:6]	0b11	3D Stereo Timing Support	(0b11 : Only Timing Code Listed as part of the block)
		[5:3]	0b000	RESERVED	
		[2:0]	0b001	Revision '1'	
02h	0x0C	[7:0]	12	Number of Payload Bytes in Block	(12 Bytes)
03h	0x02	[7:0]	2	Number of Bytes in Stereo Interface Method	(2 Bytes)
04h	0x05	[7:0]	5	Stereo Interface Method Code	(5 = Stacked Frame)
05h	0x00	[7:0]	0	View Identity	(0 : Top is Left view and Bottom is Right view)
06h	0x04	[7:6]	0b00	Timing Code Type	(0b00 = DMT Timing Code)
		[5]	0b0	RESERVED	
		[4:0]	0b00100	Number of Timing Code	(4 Timing Code)
07h	0x44	[7:0]		1920 x 1200 @ 60 RB	(DMT ID 44 _h)
08h	0x52	[7:0]		1920 x 1080 @ 60	(DMT ID 52 _h)
09h	0x33	[7:0]		1600 x 1200 @ 60	(DMT ID 33 _h)
0Ah	0x55	[7:0]		1280 x 720 @ 60	(DMT ID 55 _h)
0Bh	0x43	[7:6]	0b01	Timing Code Type	(0b01 = CEA VIC Timing Code)
		[5]	0b0	RESERVED	

		[4:0]	0b00011	Number of Timing Code	(3 Timing Code)
0Ch	0x20	[7:0]		1920 x 1080 @ 24	(CEA VIC 32 =>20 _h)
0Dh	0x3C	[7:0]		1280 x 720 @ 24	(CEA VIC 60 =>3C _h)
0Eh	0x13	[7:0]		1280 x 720 @ 50	(CEA VIC 19 =>13 _h)
Offset	Value	Bit	Value	Description	
00h	0x10	[7:0]		Stereo Display Interface Data Block	
01h	0xC1	[7:6]	0b11	3D Stereo Timing Support	(0b11 : Only Timing Code Listed as part of the block)
		[5:3]	0b000	RESERVED	
		[2:0]	0b001	Revision '1'	
02h	0x0A	[7:0]	12	Number of Payload Bytes in Block	(10 Bytes)
03h	0x02	[7:0]	2	Number of Bytes in Stereo Interface Method	(2 Bytes)
04h	0x05	[7:0]	1	Stereo Interface Method Code	(1 : Side by Side)
05h	0x00	[7:0]	0	View Identity	(0 : Left side is Left view and Right side is Right view)
06h	0x46	[7:6]	0b01	Timing Code Type	(0b01 = CEA VIC Timing Code)
		[5]	0b0	RESERVED	
		[4:0]	0b00110	Number of Timing Code	(6 Timing Code)
07h	0x20	[7:0]		1920 x 1080 @ 24	(CEA VIC 32 =>20 _h)
08h	0x3C	[7:0]		1280 x 720 @ 24	(CEA VIC 60 =>3C _h)
09h	0x04	[7:0]		1280 x 720 @ 60	(CEA VIC 4 =>04 _h)
0Ah	0x13	[7:0]		1280 x 720 @ 50	(CEA VIC 19 =>13 _h)
0Bh	0x05	[7:0]		1920 x 1080 @ 60i	(CEA VIC 5 =>05 _h)
0Ch	0x14	[7:0]		1920 x 1080 @ 50i	(CEA VIC 20 =>14 _h)

10 Appendix F: Contributors to Previous Versions

Table 10-1: Contributors to Version 1.2

Name	Company	Contribution
Syed Athar Hussain	AMD	Task Group Vice-chair & Document Co-editor
Aric Cyr	AMD	Document Co-editor
Roger Quero	AMD	
Yoshinobu Banba	EIZO NANAO	
Bob Myers	Hewlett-Packard	
Prashant Shamaraao	IDT	
Joel Silverman	Kawasaki Microelectronics	
John Bojorquez	Nvidia	
Craig Wiley	Parade Technologies	Task Group Chair
Katsuhiro Shimizu	Sony	

Table 10-2: Contributors to Version 1.1

Name	Company	Contribution
Syed Athar Hussain	AMD	Task Group Editor
Joe Goodart	Dell	
James R. Webb	Luxtera	Task Group Vice-chair
Bob Myers	Hewlett-Packard	Task Group Chair
Alain d'Hautecourt	ViewSonic	
Ashley Saldanha	Entech Taiwan	

Table 10-3: Contributors to Version 1.0

Name	Company	Contribution
Syed Athar Hussain	AMD	Task Group Editor
Joe Goodart	Dell	
James R. Webb	Granite River Labs	Task Group Vice-chair
Bob Myers	Hewlett-Packard	Task Group Chair
Chandra Konduru	Intel	
Matt Sottek	Intel	
Christian Hofmann	Kontron	
Josef Trautwein	Kontron	
David Cook	NVIDIA	
Gerrit Slavenburg	NVIDIA	
William Tsu	NVIDIA	
Glenn Adler	Philips	
Ian Miller	Miller Consulting	Member Emeritus
Robert Blanchard	Sony	
Alain d'Hautecourt	ViewSonic	