Specification for
Display Command Set (DCS^{SM})

Version 1.3

5 October 2015

MIPI Board Adopted 11 February 2016

This document is a MIPI Specification. MIPI member companies' rights and obligations apply to this MIPI Specification as defined in the MIPI Membership Agreement and MIPI Bylaws.

Further technical changes to this document are expected as work continues in the Display Working Group.

NOTICE OF DISCLAIMER

The material contained herein is not a license, either expressly or impliedly, to any IPR owned or controlled by any of the authors or developers of this material or MIPI®. The material contained herein is provided on an “AS IS” basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and MIPI hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence.

All materials contained herein are protected by copyright laws, and may not be reproduced, republished, distributed, transmitted, displayed, broadcast or otherwise exploited in any manner without the express prior written permission of MIPI Alliance. MIPI, MIPI Alliance and the dotted rainbow arch and all related trademarks, tradenames, and other intellectual property are the exclusive property of MIPI Alliance and cannot be used without its express prior written permission.

ALSO, THERE IS NO WARRANTY OF CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT. IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT OR MIPI BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT, SPECIFICATION OR DOCUMENT RELATING TO THIS MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Without limiting the generality of this Disclaimer stated above, the user of the contents of this Document is further notified that MIPI: (a) does not evaluate, test or verify the accuracy, soundness or credibility of the contents of this Document; (b) does not monitor or enforce compliance with the contents of this Document; and (c) does not certify, test, or in any manner investigate products or services or any claims of compliance with the contents of this Document. The use or implementation of the contents of this Document may involve or require the use of intellectual property rights (“IPR”) including (but not limited to) patents, patent applications, or copyrights owned by one or more parties, whether or not Members of MIPI. MIPI does not make any search or investigation for IPR, nor does MIPI require or request the disclosure of any IPR or claims of IPR as respects the contents of this Document or otherwise.

Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to:

MIPI Alliance, Inc.
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08854
Attn: Board Secretary

Contents

Figures

Tables

Release History

Date	Release	Description
2005-02-15	v1.00a	Initial MIPI Alliance Board-approved release.
2006-06-22	v1.01.00	Minor update with editorial corrections, reference updates and several bit definitions added to commands for image manipulation.
2010-10-20	v1.02.00	Minor updates containing technical clarifications and editorial updates.
2012-04-06	v1.1	Board-approved release. Added support for Stereoscopic Display Formats.
2014-06-16	v1.2	Board-approved release. Added support for command mode and display stream compression.
2016-02-11	v1.3	Board-adopted release. Added commands to access and display backlight, advanced power management, error status, and compressed image testability functions.

1 Introduction

This document defines display module behavior for devices that adhere to MIPI Specifications for mobile device host processor, and display interfaces in an abstract, device independent way. All commands in this Specification, except those indicated as optional, shall be supported by display modules that adhere to *MIPI Alliance Standard for Display Pixel Interface* [MIPI01], *MIPI Alliance Standard for Display Bus Interface* [MIPI02], and *MIPI Alliance Specification for Display Serial Interface* [MIPI03] except as provided for in the individual Specifications. Stereoscopic image support is defined in *MIPI Alliance Specification for Stereoscopic Display Formats* [MIPI05].

1.1 Scope

Display commands and logical flow are within the scope of this document. In addition, to support device abstraction, several display architectures are also specified.

Electrical specifications and interface protocols are out of scope for this document.

1.2 Purpose

This document is used by manufacturers to design products that adhere to MIPI Specifications for mobile device host processor and display interfaces.

Implementing the DCS Specification reduces the time-to-market and design cost of mobile devices by simplifying the interconnection of products from different manufacturers. In addition, adding new features such as larger or additional displays to mobile devices is simplified due to the extensible nature of MIPI Specifications.

2 Terminology

The MIPI Alliance has adopted Section 13.1 of the *IEEE Standards Style Manual*, which dictates use of the words “shall”, “should”, “may”, and “can” in the development of documentation, as follows:

> The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).
>
> The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; must is used only to describe unavoidable situations.
>
> The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.
>
> The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).
>
> The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted*).
>
> The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

All sections are normative, unless they are explicitly indicated to be informative.

2.1 Glossary

2D Mode: An operating state in which a stereoscopic-capable display is rendering one image per frame to both eyes and does not create a stereoscopic effect.

3D Mode: An operating state in which a stereoscopic-capable display renders a stereoscopic image with a unique view for each eye.

Bitstream: The sequence of data bytes resulting from the coding of image data. The bit stream does not contain a header or syntax markers.

Codestream: A sequence of data bytes composed of a bitstream and any header and syntax markers necessary for decoding. The codestream boundary usually coincides with a frame boundary, but does not need to do so.

Compressed Data: A sequence of data bytes composed of a bitstream and any header and syntax markers necessary for decoding.

Display Area: The portion of a display device used to show image data.

Display Controller: A separate silicon chip, or integrated functional block in a host device, used to control a display module. May include full-frame or partial-frame memory.

Display Device: A functional device that shows images such as a Liquid Crystal Display.

Display Driver: An integrated circuit inside a display module used to control the display device. May or may not integrate full or partial frame-memory.

Display Glass: Same as Display Device. Derived from the display material’s name.

Display Module: A functional module used to show an image. Can consist of a display device, display driver, additional peripheral components or circuits and a display interface.

Display Panel: Same as Display Device.

Frame Memory: Memory integrated in a display driver or display controller in order to provide storage for display device refreshment. Full-frame memory provides enough storage for the full display area of a display device. Partial-frame memory provides only enough storage for a portion of the display area.

Frame-based: The data transfer mode that sends an entire left or right view followed by the corresponding right or left view, respectively.

Frame-sequential: Same as Temporal Mode.

Landscape: The horizontal dimension exceeds the vertical dimension. If square, defined by the manufacturer.

Landscape Scanning: The pixel writing direction from the display driver to the display in which the number of pixels written per line exceeds the number of lines.

Landscape/Portrait Orientation: The orientation the display is viewed by a user.

Landscape/Portrait Switchable: A display where the stereoscopic effect can be switched between landscape and portrait orientation.

Left View: Part of the stereoscopic image intended to be viewed by the user’s left eye.

Left-Right Order: This value defines whether the first pixel, line, or frame of 3D Mode content sent across the physical link is intended for viewing by the left eye or the right eye. The order may apply with respect to pixel-based, line-based or frame-based modes of transmission

Line-based: The data transfer mode that sends an entire left or right line followed by the corresponding right or left line, respectively.

Portrait: The vertical dimension exceeds the horizontal dimension. If square, defined by the manufacturer.

Portrait Scanning: The pixel writing direction from the display driver to the display in which the number of lines written exceeds the number of pixels per line.

Right View: Part of the stereoscopic image intended to be viewed by the user’s right eye.

Spatial: The left and right views are shown simultaneously to the viewer.

Stereoscopic Image: A pair of offset images of a scene (views) that renders content to both the left eye and right eye to produce the perception of depth.

Temporal Mode: A time-sequential stereoscopic image in which the left view and right view are alternately presented to the user and directed to the appropriate eye.

Type 1 Display Architecture: A display module architecture in which the display module includes a display device, display driver, full-frame memory, interface registers, timing controller, non-volatile memory and a control interface.

Type 2 Display Architecture: A display module architecture in which the display module includes a display device, display driver, partial-frame memory, interface registers, timing controller, non-volatile memory, a control interface and a video stream interface.

Type 3 Display Architecture: Similar to the Type 2 Display Architecture except no frame memory is present.

2.2 Abbreviations

e.g.	For example (Latin: exempli gratia)
i.e.	That is (Latin: id est)

2.3 Acronyms

DBI	Display Bus Interface

DCS	Display Command Set
DPI	Display Pixel Interface
DSI	Display Serial Interface

3 References

[MIPI01] *MIPI Alliance Standard for Display Pixel Interface (DPI-2)*, version 2.00, MIPI Alliance, Inc., 15 September 2005.

[MIPI02] *MIPI Alliance Standard for Display Bus Interface (DBI-2)*, version 2.00, MIPI Alliance, Inc., 29 November 2005.

[MIPI03] *MIPI Alliance Specification for Display Serial Interface (DSI)*, version 1.3, MIPI Alliance, Inc., 10 March 2015.

[MIPI04] *MIPI Alliance Specification for Device Descriptor Block (DDB)*, version 1.0, MIPI Alliance, Inc., 29 October 2008.

[MIPI05] *MIPI Alliance Specification for Stereoscopic Display Formats (SDF)*, version 1.0, MIPI Alliance, Inc., 14 March 2012.

[VESA01] *Display Stream Compression Standard*, version 1.1, VESA, www.vesa.org, 1 August 2014.

4 Display Architectures

The display module shall be based on Type 1, Type 2 or Type 3 display architecture.

The Type 1 Display Architecture should consist of the following functional blocks:

Display Device. The Display Device is used to show image data.

Display Driver. The Display Driver may be one or more devices used to drive the display device.

Frame memory. Frame Memory holds compressed or uncompressed image data depending upon whether compression is required for the display or not. Frame memory can be integrated in the display driver.

Registers. Registers are used to configure display behavior and identification information. Registers can be integrated in the display driver.

Timing Controller. The Timing Controller provides timing signals to control the display and display driver based on configuration information. The Timing Controller can be integrated in the display driver.

Non-volatile Memory. Non-volatile Memory is used to store default register and configuration values. Non-volatile memory can be integrated in the display driver.

Control Interface. The Control Interface is the interface between the host processor and the display driver. The Control Interface can be integrated in the display driver.

Display Driving Circuit. The Display Driving Circuit converts timing signals and voltages to signals appropriate to drive the display device.

Decoder (optional). The Decoder decodes compressed data from the host processor and generates pixel data to pass to the display device. The decoder block is optional as compression is dependent upon system requirements. The Decoder can be integrated in the display driver.

Power Supply. The Power Supply converts system voltages to levels usable by the display device and display driver. The Power Supply can be integrated in the display driver.

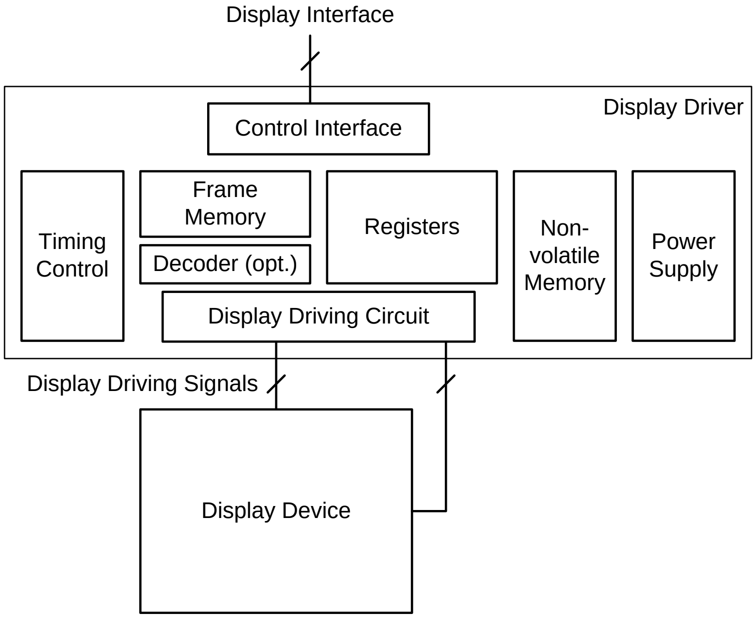


Figure 1 Type 1 Display Architecture Block Diagram

The Type 2 Display Architecture should consist of the following functional blocks:

Display Device. The Display Device is used to show image data.

Display Driver. The Display Driver may be one or more devices used to drive the display device.

Partial-frame Memory. Partial-frame Memory holds compressed or uncompressed image data depending upon whether compression is required for the display or not. Partial-frame memory can be integrated in the display driver.

Registers. Registers are used to configure display behavior and identification information. Registers can be integrated in the display driver.

Timing Controller. The Timing Controller provides timing signals to control the display and display driver based on configuration information. The Timing Controller can be integrated in the display driver.

Non-volatile memory. Non-volatile Memory is used to store default register and configuration values. Can be integrated in the display driver.

Control Interface. The Control Interface is the interface between the host processor and the display driver. The Control Interface can be integrated in the display driver.

Display Driving Circuit. The Display Driving Circuit converts timing signals and voltages to signals appropriate to drive the display device.

Decoder (optional). The Decoder decodes compressed data from the host processor and generates pixel data to pass to the display device. The decoder block is optional as compression is dependent upon system requirements. The Decoder can be integrated in the display driver.

Power Supply. The Power Supply converts system voltages to levels usable by the display device and display driver. The Power Supply can be integrated in the display driver.

Video Stream Interface. The Video Stream Interface receives video image data and timing signals from the host processor.

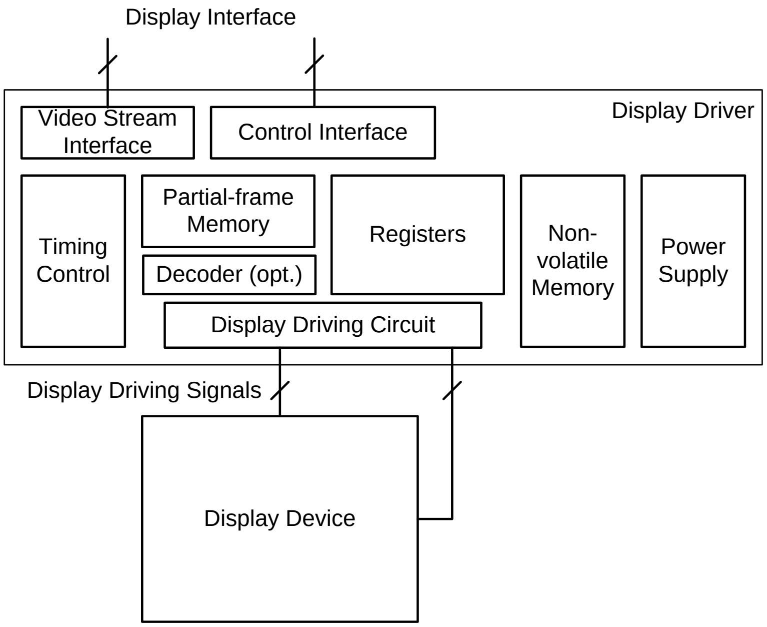


Figure 2 Type 2 Display Architecture Block Diagram

The Type 3 Display Architecture should consist of the following functional blocks:

Display Device. The Display Device is used to show image data.

Display Driver. The Display Driver may be one or more devices used to drive the display device.

Registers. Registers are used to configure display behavior and identification information. Registers can be integrated in the display driver.

Timing Controller. The Timing Controller provides timing signals to control the display and display driver based on configuration information. The Timing Controller can be integrated in the display driver.

Non-volatile memory. Non-volatile Memory is used to store default register and configuration values. Can be integrated in the display driver.

Control Interface. The Control Interface is the interface between the host processor and the display driver. The Control Interface can be integrated in the display driver.

Display Driving Circuit. The Display Driving Circuit converts timing signals and voltages to signals appropriate to drive the display device.

Decoder (optional). The Decoder decodes compressed data from the host processor and generates pixel data to pass to the display device. The decoder block is optional as compression is dependent upon system requirements. The Decoder can be integrated in the display driver.

Power Supply. The Power Supply converts system voltages to levels usable by the display device and display driver. The Power Supply can be integrated in the display driver.

Video Stream Interface. The Video Stream Interface receives video image data and timing signals from the host processor.

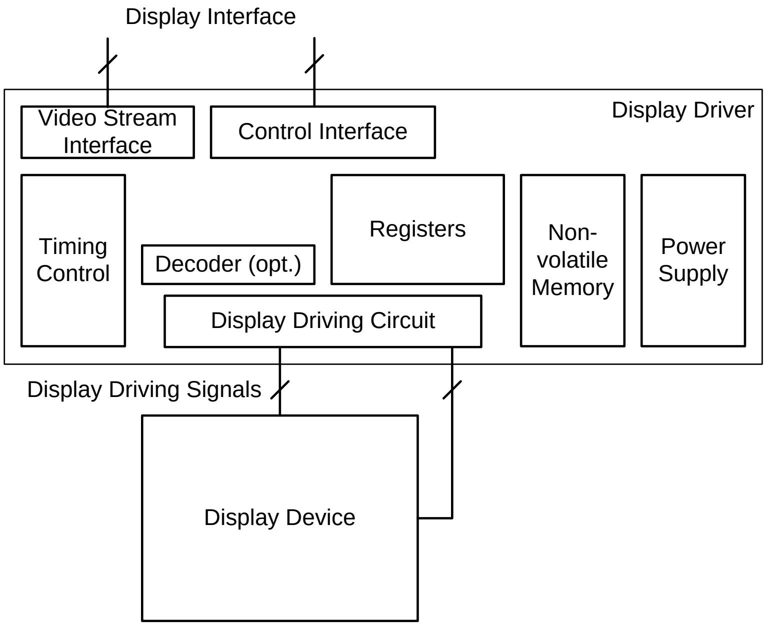


Figure 3 Type 3 Display Architecture Block Diagram

In all architecture types, it is assumed the power supply is under the control of the display driver.

The Display Command Set is used through the mentioned control interface.

5 Display Functional Description

5.1 Power Level Definition

A display module designed using the Type 1 display architecture shall implement the power sequence shown in Figure 4.

A display module designed using the Type 2 display architecture shall implement the power sequence shown in Figure 5.

A display module designed using the Type 3 display architecture shall implement the power sequence shown in Figure 6.

Each power sequence consists of a combination of different display and power modes as follows.

In Normal mode, the display module shows image data using the full display area of the display device. See Section 6.3 for a description of Normal mode.

In Partial mode, the display module shows image data in only a portion of the full display area of the display device. See Section 6.44 for a description of Partial mode.

In Idle mode, the display module shows image data using a limited number of colors. Turning off Idle mode displays the image data using the full number of colors supported by the display device. See Section 6.1 for a description of Idle mode.

In Sleep mode, the display module does not show any image data. In addition, the display interface shall remain powered and along with those functional blocks necessary to maintain the data in the frame memory and registers. The remaining functional blocks are placed in their low power modes. See Section 6.5 for a description of Sleep mode.

When Sleep mode is off, the display module shows image data on the display device and all functional blocks operate normally. See Section 6.8 for a description of operation when Sleep mode is off.

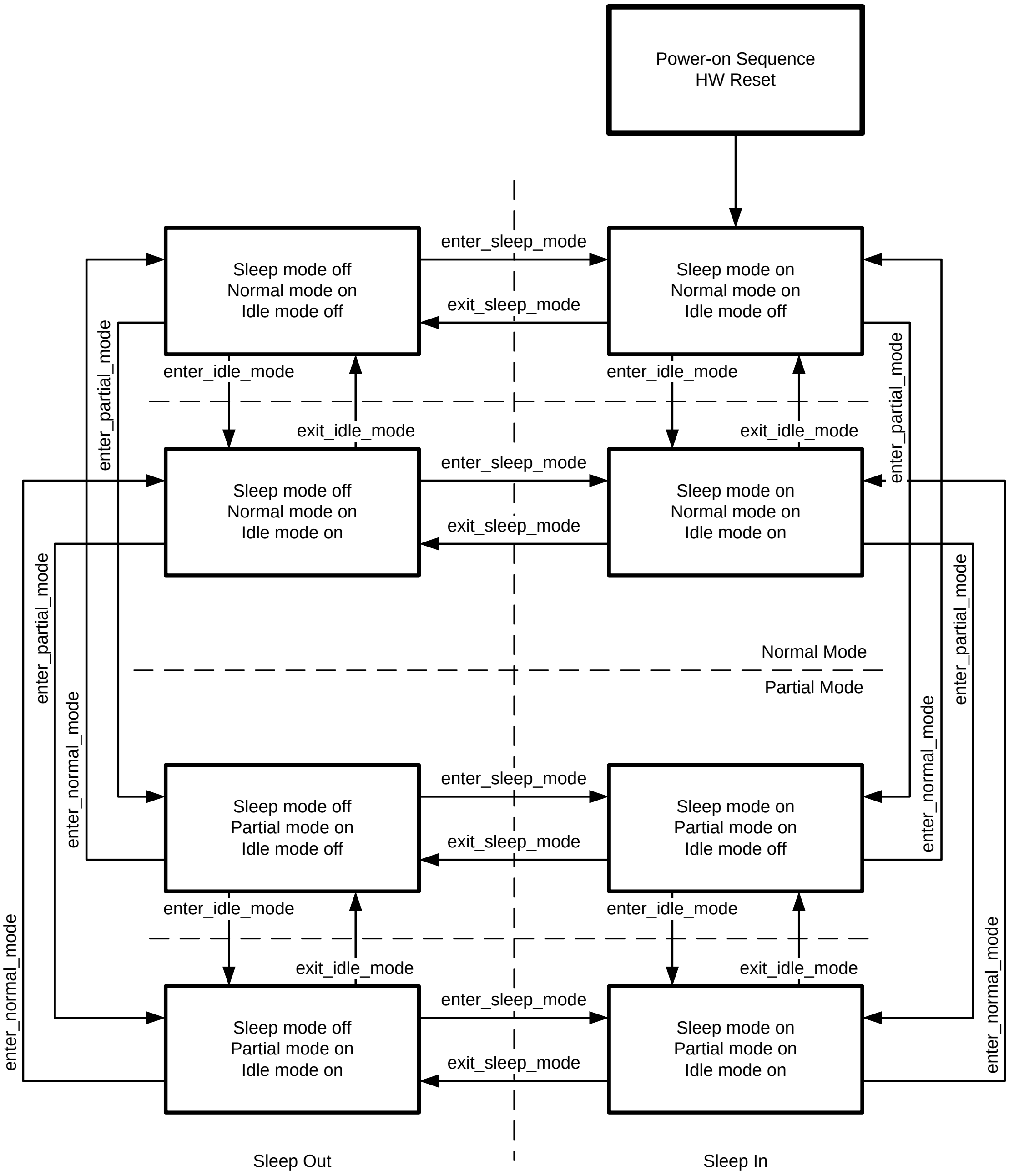


Figure 4 Type 1 Display Architecture Power Change Sequences

Note 1: There shall be no abnormal visual effect when changing between power modes.

Note 2: The display module can change between any power modes without restriction.

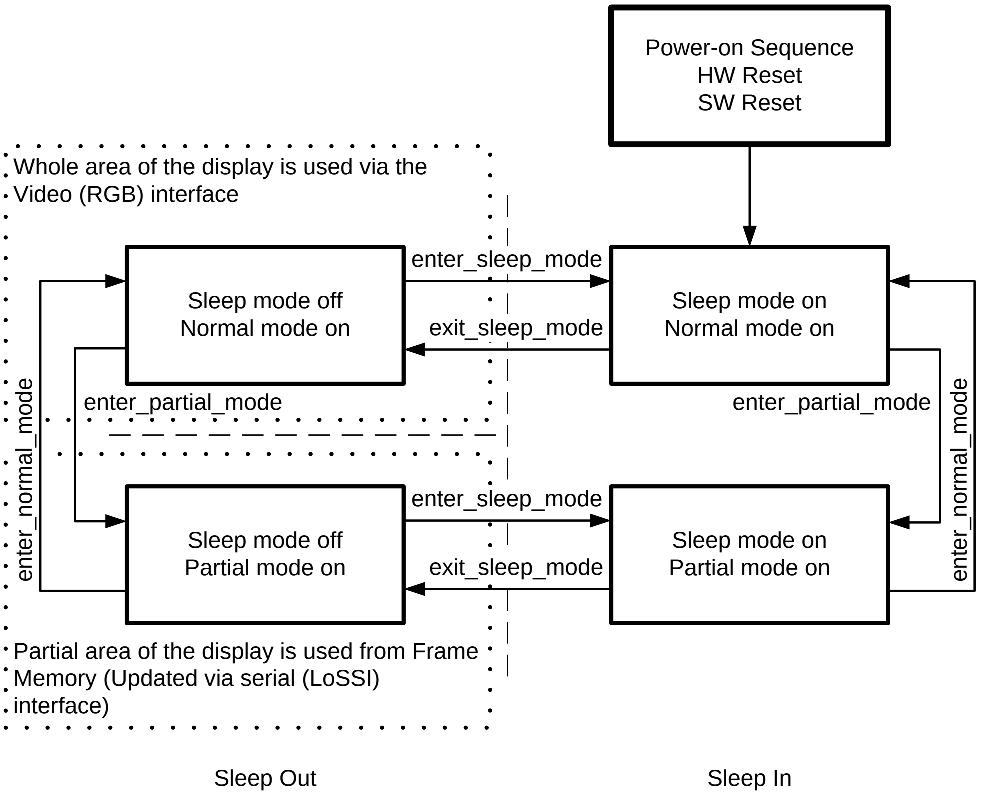


Figure 5 Type 2 Display Architecture Power Change Sequence

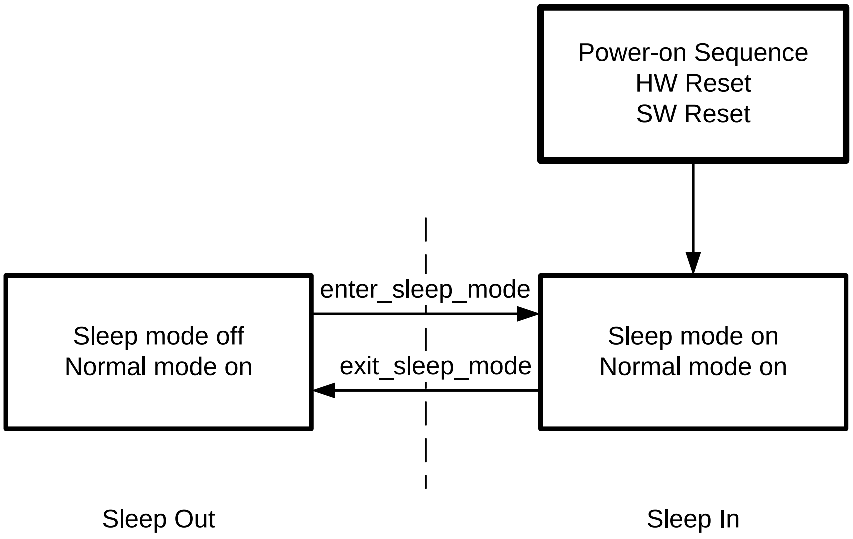


Figure 6 Type 3 Display Architecture Power Change Sequence

Note 1: There shall be no abnormal visual effect when changing between power modes.

Note 2: The display module can change between any power modes without restriction.

5.2 Gamma Curves

The display module can implement a gamma adjustment. If gamma adjustment is implemented then the display module shall support at a minimum Gamma Curve 1 as described in Section 5.2.1. The display module can also implement up to three additional gamma curves as described in Section 5.2.2 through Section 5.2.4. The gamma curve is selected by the set_gamma_curve command, as described in Section 6.42.

In the gamma curve figures, **x** is the normalized image data supplied by the host processor to the display module and **y** is the normalized response of the display device.

5.2.1 Gamma Curve 1 (GC0)

Gamma Curve 1 (GC0) is 2.2, i.e. $y=x^{2.2}$

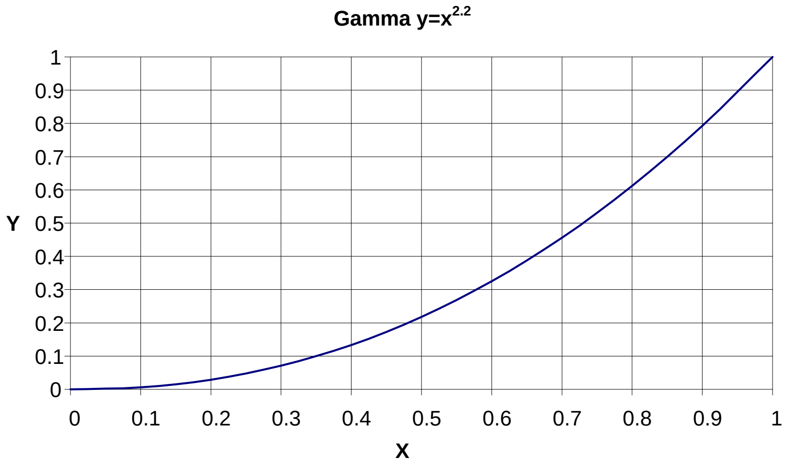


Figure 7 Gamma curve 1 (GC0)

5.2.2 Gamma Curve 2 (GC1)

Gamma Curve 2 (GC1) is 1.8, i.e. $y=x^{1.8}$

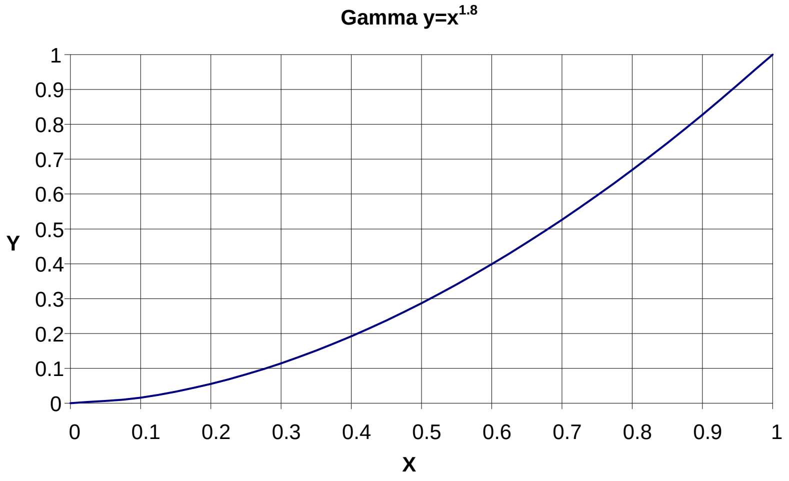


Figure 8 Gamma Curve 2 (GC1)

5.2.3 Gamma Curve 3 (GC2)

Gamma Curve 3 (GC2) is 2.5, i.e. $y=x^{2.5}$

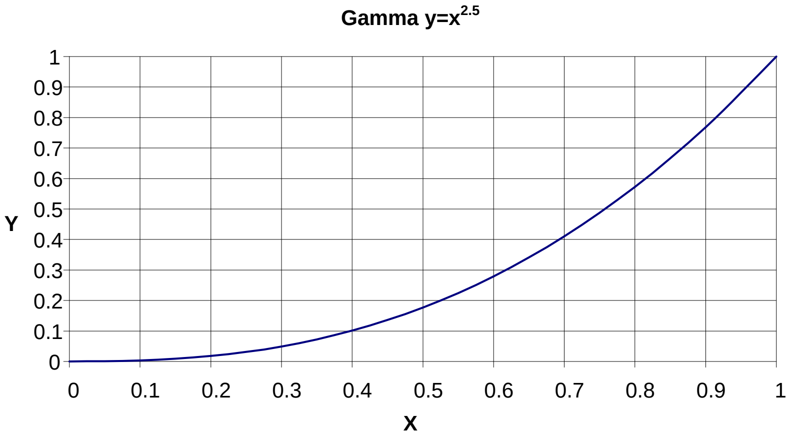


Figure 9 Gamma Curve 3 (GC2)

5.2.4 Gamma Curve 4 (GC3)

Gamma Curve 4 (GC3) is linear, i.e. $y=x^1$

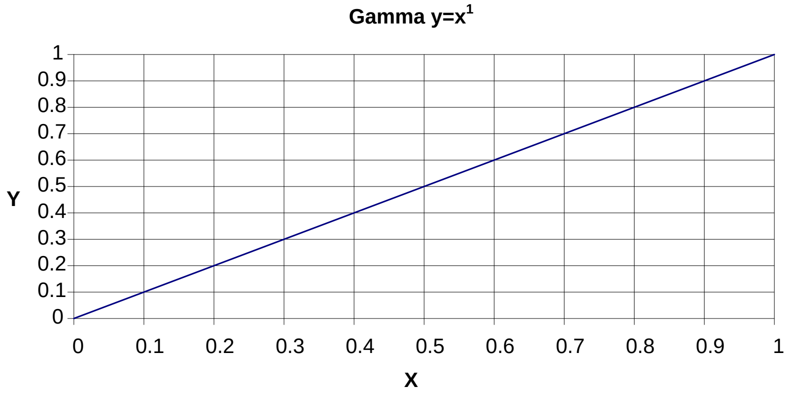


Figure 10 Gamma Curve 4 (GC3)

5.3 Self-diagnostic Functions

The display module shall support all the self-diagnostic functions in this section except those functions indicated as optional. Optional functions can be implemented in the display module at the manufacturer's discretion.

5.3.1 Register Loading Detection

The exit_sleep_mode command (see Section 6.8) is a trigger for the Register Loading Detection function. This function indicates if the display module correctly loaded the factory default values from Non-volatile memory to the registers. If the registers were loaded properly then bit D7 of the SDR register is inverted, otherwise the value is unchanged. See Section 6.15 for a description of the SDR register.

The flow chart for the Register Loading Detection function is shown in Figure 11.

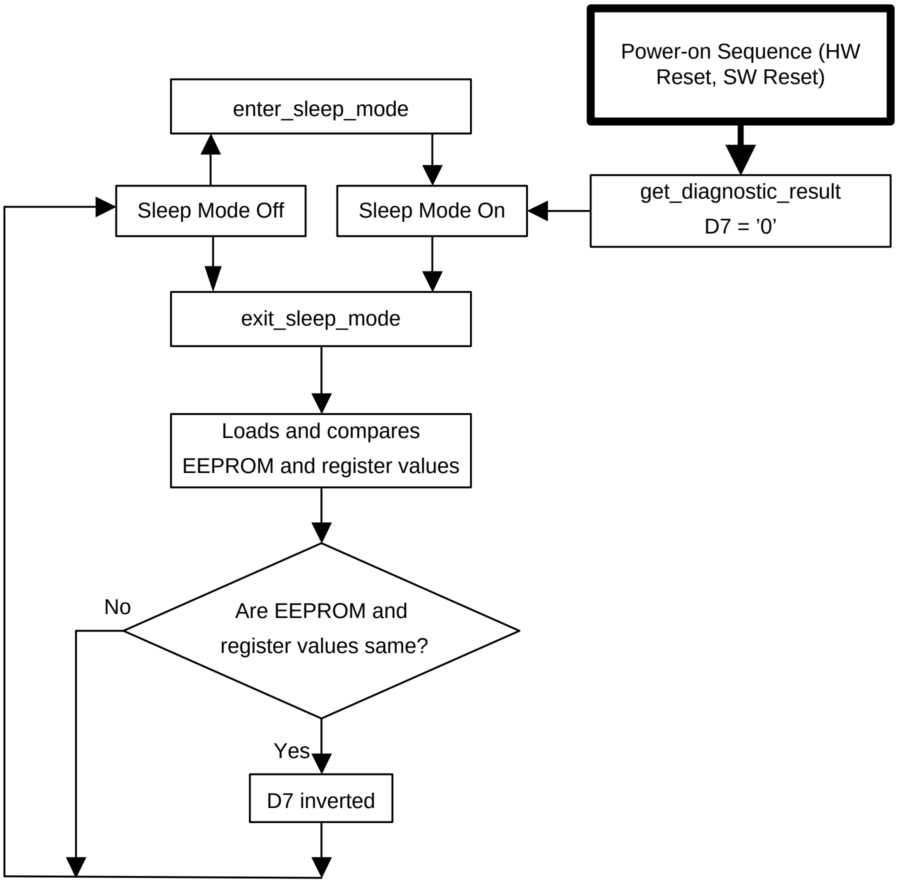

Power-on Sequence (HW Reset, SW Reset)
enter_sleep_mode
Sleep Mode Off
Sleep Mode On
get_diagnostic_result
D7 = '0'
exit_sleep_mode
Loads and compares EEPROM and register values
Are EEPROM and register values same?
No
Yes
D7 inverted

Figure 11 Register Loading Detection Flow Chart

Note:

Registers modified by the display module after loading are not verified.

5.3.2 Functionality Detection

The exit_sleep_mode command (see Section 6.8) is a trigger for the Functionality Detection function. This function indicates if the display module functional blocks, e.g. power supply, clock generator, etc. are operating correctly. If the functional blocks are operating properly then bit D6 of the SDR register is inverted, otherwise the value is unchanged. See Section 6.15 for a description of the SDR register.

The flow chart for the Functionality Detection function is shown in Figure 12.

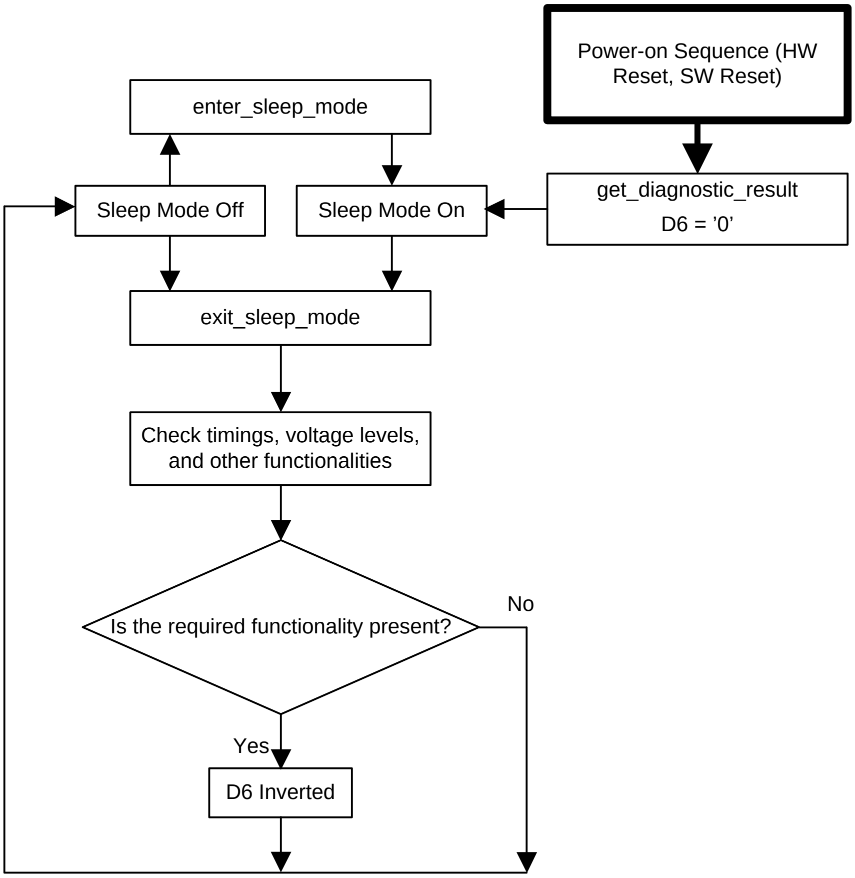

Power-on Sequence (HW Reset, SW Reset)
enter_sleep_mode
Sleep Mode Off
Sleep Mode On
get_diagnostic_result
D6 = ’0’
exit_sleep_mode
Check timings, voltage levels, and other functionalities
Is the required functionality present?
No
Yes
D6 Inverted

Figure 12 Functionality Detection Flow Chart

The host processor shall wait before sending a get_power_mode command so the display module can exit Sleep mode and finish the Functionality Detection function.

5.3.3 Chip Attachment Detection (optional)

The exit_sleep_mode command (see Section 6.8) is a trigger for the Chip Attachment Detection function. This function indicates if certain chips, e.g. display driver IC, are attached to the display module. If the chips are properly attached to the display module then bit D5 of the SDR register is inverted, otherwise the value is unchanged. See Section 6.15 for a description of the SDR register.

The flow chart for the Chip Attachment Detection function is shown in Figure 14.

Figure 13 is a reference implementation for the Chip Attachment Detection function. Two bumps are connected together via a conductor on the flex foil or the display glass substrate in all four corners of the chip.

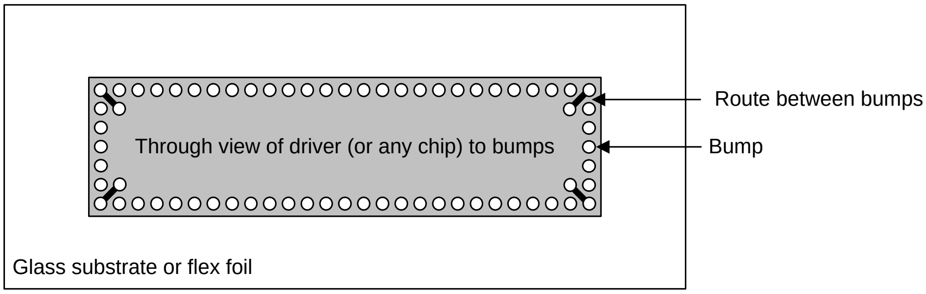

Route between bumps
Bump
Through view of driver (or any chip) to bumps
Glass substrate or flex foil

Figure 13 Chip Attachment Detection Reference

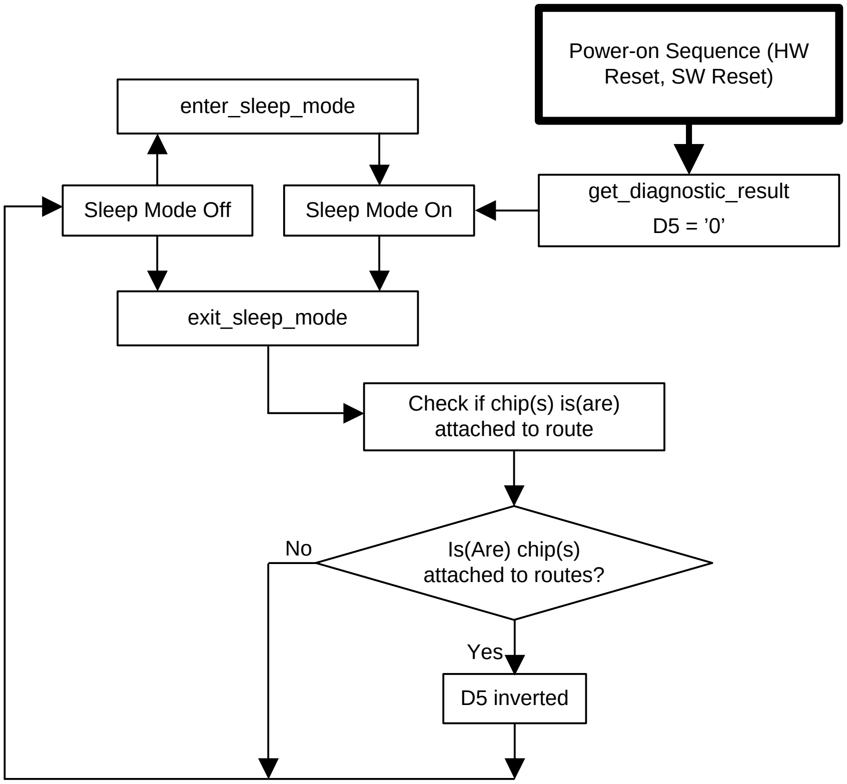

Power-on Sequence (HW Reset, SW Reset)
enter_sleep_mode
Sleep Mode Off
Sleep Mode On
get_diagnostic_result
D5 = '0'
exit_sleep_mode
Check if chip(s) is(are) attached to route
No
Is(Are) chip(s) attached to routes?
Yes
D5 inverted

Figure 14 Chip Attachment Detection Flow Chart

5.3.4 Display Glass Break Detection (optional)

The exit_sleep_mode command (see Section 6.8) is a trigger for the Display Glass Break Detection function. This function indicates if display glass is broken. If the display glass is broken then bit D4 of the SDR register is inverted, otherwise the value is unchanged. See Section 6.15 for a description of the SDR register.

The flow chart for the Display Glass Break Detection function is shown in Figure 16.

Figure 15 is a reference implementation for the Display Glass Break Detection function. Two bumps are connected together via a conductor routed on the outside edge of the display glass substrate.

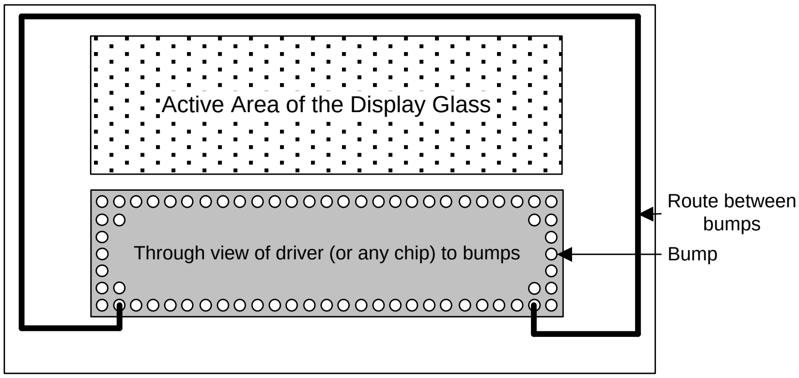

Active Area of the Display Glass
Through view of driver (or any chip) to bumps
Route between bumps
Bump

Figure 15 Display Glass Break Detection Reference

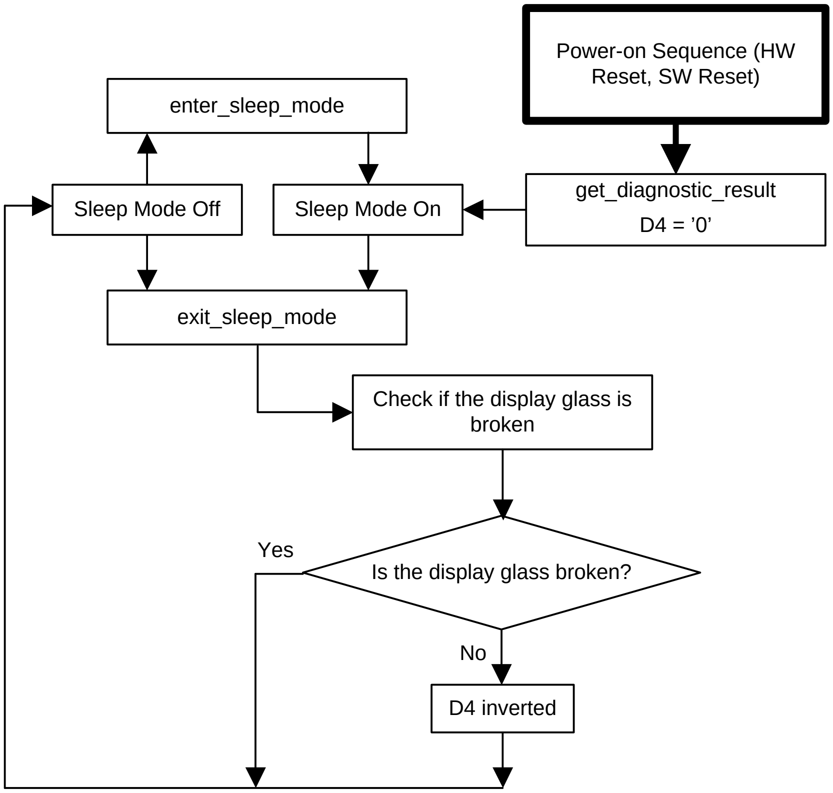


Figure 16 Display Glass Break Detection Flow Chart

5.4 Display Command Set

The Display Command Set is used to store image data, configure the display module behavior and retrieve display module data including identification information by accessing the frame memory and the display module registers.

The DCS is separated into two functional areas: the User Command Set and the Manufacturer Command Set. Each command is an eight-bit code with 00h to AFh assigned to the User Command Set and all other codes assigned to the Manufacturer Command Set.

The Manufacturer Command Set (MCS) is a device dependent interface intended for factory programming of the display module default parameters. Once the display module is configured, the MCS shall be disabled by the manufacturer. Once disabled, all MCS commands are ignored by the display interface. The MCS is out of scope for this document.

The User Command Set provides a display device independent interface targeted at the operating system's hardware abstraction layer. All commands listed in this section shall be implemented except write_LUT, get_3D_control, set_3D_control and get_compression_mode, which are optional.

Any unused command codes shall be ignored by the display module.

The remainder of this section is divided into three sections. Section 5.5 is an alphabetical list of the supported commands. Section 5.6 and Section 5.7 describe command functionality in different display architectures and operating modes.

5.5 Command Lists

Table 1 lists all DCS Commands in alphabetical order, and Table 2 lists them in Hex Code order.

Table 1 Command List – Alphabetical

Command	Hex Code	Description	Number of Parameters	Display Architecture Implementation Requirement		
				Type 1	Type 2	Type 3
enter_idle_mode	39h	Reduced color depth is used on the display panel.	0	Yes	No	No
enter_invert_mode	21h	Displayed image colors are inverted.	0	Yes	Yes	Yes
enter_normal_mode	13h	The whole display area is used for image display.	0	Yes	Yes	No
enter_partial_mode	12h	Part of the display area is used for image display.	0	Yes	Yes	No
enter_sleep_mode	10h	Power for the display panel is off.	0	Yes	Yes	Yes
exit_idle_mode	38h	Full color depth is used on the display panel.	0	Yes	No	No
exit_invert_mode	20h	Displayed image colors are not inverted.	0	Yes	Yes	Yes
exit_sleep_mode	11h	Power for the display panel is on.	0	Yes	Yes	Yes
get_3D_control	3Fh	Get display module 3D Mode.	2	No	No	No
get_address_mode	0Bh	Get the data order for transfers from the Host to the display module and from the frame memory to the display device.	1	Yes	Yes	Yes
get_blue_channel	08h	Get the blue component of the pixel at (0, 0).	1	No	Yes	Yes
get_CABC_min_brightness	5Fh	Get current minimum brightness level of the active CABC mode	1 or 2	Yes	Yes	Yes
get_compression_mode	03h	Get the current compression mode	1	No	No	No
get_control_display	54h	Get control display mode	1	Yes	Yes	Yes
get_diagnostic_result	0Fh	Get Peripheral Self-Diagnostic Result	1	Yes	Yes	Yes
get_display_brightness	52h	Get current display brightness level	1 or 2	Yes	Yes	Yes
get_display_mode	0Dh	Get the current display mode from the peripheral.	1	Yes	Yes	Yes
get_error_count_on_DSI	05h	Get number of corrupted packets on DSI	1	Yes	Yes	Yes

Command	Hex Code	Description	Number of Parameters	Display Architecture Implementation Requirement		
				Type 1	Type 2	Type 3
get_green_channel	07h	Get the green component of the pixel at (0, 0).	1	No	Yes	Yes
get_image_checksum_ct	15h	Returns the checksum of a frame of color-transformed pixel data	2	Yes*	Yes*	Yes*
get_image_checksum_rgb	14h	Returns the checksum of a frame of RGB pixel data	2	Yes*	Yes*	Yes*
get_pixel_format	0Ch	Get the current pixel format.	1	Yes	Yes	Yes
get_power_mode	0Ah	Get the current power mode.	1	Yes	Yes	Yes
get_power_save	56h	Get power save mode	1	Yes	Yes	Yes
get_red_channel	06h	Get the red component of the pixel at (0, 0).	1	No	Yes	Yes
get_scanline	45h	Get the current scanline.	2	Yes	Yes	No
get_signal_mode	0Eh	Get display module signaling mode.	1	Yes	Yes	Yes
nop	00h	No Operation	0	Yes	Yes	Yes
read_DDB_continue	A8h	Continue reading the DDB from the last read location.	variable	Yes	Yes	Yes
read_DDB_start	A1h	Read the DDB from the provided location.	variable	Yes	Yes	Yes
read_memory_continue	3Eh	Read image data from the peripheral continuing after the last read_memory_continue or read_memory_start.	variable	Yes	Yes	No
read_memory_start	2Eh	Transfer image data from the peripheral to the Host Processor interface starting at the location provided by set_column_address and set_page_address.	variable	Yes	Yes	No
read_PPS_continue	A9h	Continue reading the specified length of PPS data immediately following the last read location.	variable	Yes	Yes	Yes
read_PPS_start	A2h	Transfer the specified length of PPS data starting at the beginning of the PPS register.	variable	Yes	Yes	Yes
set_3D_control	3Dh	3D is used on the display panel.	2	No	No	No
set_address_mode	36h	Set the data order for transfers from the Host to the display module and from the frame memory to the display device.	1	Yes	Yes	Yes

Command	Hex Code	Description	Number of Parameters	Display Architecture Implementation Requirement		
				Type 1	Type 2	Type 3
set_CABC_min_brightness	5Eh	Writes minimum brightness level for the active CABC mode.	1 or 2	Yes	Yes	Yes
set_column_address	2Ah	Set the column extent.	4	Yes	Yes	No
set_display_brightness	51h	Selects display brightness level	1 or 2	Yes	Yes	Yes
set_display_off	28h	Blanks the display device.	0	Yes	Yes	Yes
set_display_on	29h	Show the image on the display device.	0	Yes	Yes	Yes
set_gamma_curve	26h	Selects the gamma curve used by the display device.	1	Yes	Yes	Yes
set_page_address	2Bh	Set the page extent.	4	Yes	Yes	No
set_partial_columns	31h	Defines the number of columns in the partial display area on the display device.	4	Yes	Yes	No
set_partial_rows	30h	Defines the number of rows in the partial display area on the display device.	4	Yes	Yes	No
set_pixel_format	3Ah	Defines how many bits per pixel are used in the interface.	1	Yes	Yes	Yes
set_scroll_area	33h	Defines the vertical scrolling and fixed area on display device.	6	Yes	No	No
set_scroll_start	37h	Defines the vertical scrolling starting point.	2	Yes	No	No
set_tear_off	34h	Synchronization information is not sent from the display module to the host processor.	0	Yes	No	No
set_tear_on	35h	Synchronization information is sent from the display module to the host processor at the start of VFP.	1	Yes	No	No
set_tear_scanline	44h	Synchronization information is sent from the display module to the host processor when the display device refresh reaches the provided scanline.	2	Yes	No	No
set_vsync_timing	40h	Set VSYNC timing	1	No	No	No
soft_reset	01h	Software Reset	0	Yes	Yes	Yes
write_control_display	53h	Writes control mode of display brightness	1	Yes	Yes	Yes

Command	**Hex Code**	**Description**	**Number of Parameters**	**Display Architecture Implementation Requirement**		
				Type 1	Type 2	Type 3
write_LUT	2Dh	Fills the peripheral look-up table with the provided data.	variable	optional	No	No
write_memory_continue	3Ch	Transfer image information from the Host Processor interface to the peripheral from the last written location.	variable	Yes	Yes	No
write_memory_start	2Ch	Transfer image data from the Host Processor to the peripheral starting at the location provided by set_column_address and set_page_address.	variable	Yes	Yes	No
write_power_save	55h	Writes power save mode	1	Yes	Yes	Yes

** Required only when the display module contains an optional bitstream decoder*

Table 2 Command List – By Hex Code

Hex Code	Command
00h	nop
01h	soft_reset
03h	get_compression_mode
05h	get_error_count_on_DSI
06h	get_red_channel
07h	get_green_channel
08h	get_blue_channel
0Ah	get_power_mode
0Bh	get_address_mode
0Ch	get_pixel_format
0Dh	get_display_mode
0Eh	get_signal_mode
0Fh	get_diagnostic_result
10h	enter_sleep_mode
11h	exit_sleep_mode
12h	enter_partial_mode
13h	enter_normal_mode
14h	get_image_checksum_rgb
15h	get_image_checksum_ct
20h	exit_invert_mode
21h	enter_invert_mode
26h	set_gamma_curve
28h	set_display_off
29h	set_display_on
2Ah	set_column_address
2Bh	set_page_address
2Ch	write_memory_start
2Dh	write_LUT
2Eh	read_memory_start
30h	set_partial_rows
31h	set_partial_columns
33h	set_scroll_area
34h	set_tear_off
35h	set_tear_on
36h	set_address_mode
37h	set_scroll_start
38h	exit_idle_mode
39h	enter_idle_mode
3Ah	set_pixel_format
3Ch	write_memory_continue
3Dh	set_3D_control
3Eh	read_memory_continue
3Fh	get_3D_control
40h	set_vsync_timing
44h	set_tear_scanline
45h	get_scanline
51h	set_display_brightness
52h	get_display_brightness
53h	write_control_display
54h	get_control_display
55h	write_power_save
56h	get_power_save
5Eh	set_CABC_min_brightness
5Fh	get_CABC_min_brightness
A1h	read_DDB_start
A2h	read_PPS_start
A8h	read_DDB_continue
A9h	read_PPS_continue

5.6 Command Accessibility

Table 3 provides command accessibility of several combinations of display and power modes.

Table 3 Command Accessibility

Command	Hex Code	Command Accessibility				
		Normal Mode On, Idle Mode Off, Sleep Mode Off	Normal Mode On, Idle Mode On, Sleep Mode Off	Partial Mode On, Idle Mode Off, Sleep Mode Off	Partial Mode On, Idle Mode On, Sleep Mode Off	Sleep Mode On
enter_idle_mode	39h	Yes	Yes	Yes	Yes	Yes
enter_invert_mode	21h	Yes	Yes	Yes	Yes	Yes
enter_normal_mode	13h	Yes	Yes	Yes	Yes	Yes
enter_partial_mode	12h	Yes	Yes	Yes	Yes	Yes
enter_sleep_mode	10h	Yes	Yes	Yes	Yes	Yes
exit_idle_mode	38h	Yes	Yes	Yes	Yes	Yes
exit_invert_mode	20h	Yes	Yes	Yes	Yes	Yes
exit_sleep_mode	11h	Yes	Yes	Yes	Yes	Yes
get_3D_control	3Fh	Yes	Yes	No	No	Yes
get_address_mode	0Bh	Yes	Yes	Yes	Yes	Yes
get_blue_channel	08h	Yes	Yes	N/A	N/A	Yes
get_CABC_min_brightness	5Fh	Yes	Yes	Yes	Yes	Yes
get_compression_mode	03h	Yes	Yes	Yes	Yes	Yes
get_control_display	54h	Yes	Yes	Yes	Yes	Yes
get_diagnostic_result	0Fh	Yes	Yes	Yes	Yes	Yes
get_display_brightness	52h	Yes	Yes	Yes	Yes	Yes
get_display_mode	0Dh	Yes	Yes	Yes	Yes	Yes
get_error_count_on_DSI	05h	Yes	Yes	Yes	Yes	Yes
get_green_channel	07h	Yes	Yes	N/A	N/A	Yes
get_image_checksum_ct	15h	Yes	Yes	No	No	No
get_image_checksum_rgb	14h	Yes	Yes	No	No	No
get_pixel_format	0Ch	Yes	Yes	Yes	Yes	Yes
get_power_mode	0Ah	Yes	Yes	Yes	Yes	Yes
get_power_save	56h	Yes	Yes	Yes	Yes	Yes
get_red_channel	06h	Yes	Yes	N/A	N/A	Yes
get_scanline	45h	Yes	Yes	Yes	Yes	Yes
get_signal_mode	0Eh	Yes	Yes	Yes	Yes	Yes
nop	00h	Yes	Yes	Yes	Yes	Yes
read_DDB_continue	A8h	Yes	Yes	Yes	Yes	Yes
read_DDB_start	A1h	Yes	Yes	Yes	Yes	Yes

Command	Hex Code	Command Accessibility				
		Normal Mode On, Idle Mode Off, Sleep Mode Off	Normal Mode On, Idle Mode On, Sleep Mode Off	Partial Mode On, Idle Mode Off, Sleep Mode Off	Partial Mode On, Idle Mode On, Sleep Mode Off	Sleep Mode On
read_memory_continue	3Eh	Yes	Yes	Yes	Yes	Yes
read_memory_start	2Eh	Yes	Yes	Yes	Yes	Yes
read_PPS_continue	A9h	Yes	Yes	Yes	Yes	Yes
read_PPS_start	A2h	Yes	Yes	Yes	Yes	Yes
set_3D_control	3Dh	Yes	Yes	No	No	Yes
set_address_mode	36h	Yes	Yes	Yes	Yes	Yes
set_CABC_min_brightness	5Eh	Yes	Yes	Yes	Yes	Yes
set_column_address	2Ah	Yes	Yes	Yes	Yes	Yes
set_display_brightness	51h	Yes	Yes	Yes	Yes	Yes
set_display_off	28h	Yes	Yes	Yes	Yes	Yes
set_display_on	29h	Yes	Yes	Yes	Yes	Yes
set_gamma_curve	26h	Yes	Yes	Yes	Yes	Yes
set_page_address	2Bh	Yes	Yes	Yes	Yes	Yes
set_partial_columns	31h	Yes	Yes	Yes	Yes	Yes
set_partial_rows	30h	Yes	Yes	Yes	Yes	Yes
set_pixel_format	3Ah	Yes	Yes	Yes	Yes	Yes
set_scroll_area	33h	Yes	Yes	Yes	Yes	Yes
set_scroll_start	37h	Yes	Yes	Yes	Yes	Yes
set_tear_off	34h	Yes	Yes	Yes	Yes	Yes
set_tear_on	35h	Yes	Yes	Yes	Yes	Yes
set_tear_scanline	44h	Yes	Yes	Yes	Yes	Yes
set_vsync_timing	40h	Yes	Yes	Yes	Yes	Yes
soft_reset	01h	Yes	Yes	Yes	Yes	Yes
write_control_display	53h	Yes	Yes	Yes	Yes	Yes
write_LUT	2Dh	Yes	Yes	Yes	Yes	Yes
write_memory_continue	3Ch	Yes	Yes	Yes	Yes	Yes
write_memory_start	2Ch	Yes	Yes	Yes	Yes	Yes
write_power_save	55h	Yes	Yes	Yes	Yes	Yes

5.7 Default Modes and Values

Table 4 provides default display modes, power modes and register values.

Table 4 Default Display Mode, Power Mode and Register Values

Command	Hex Code	Parameters	Default Modes and Values, Hex		
			Power-on Sequence	SW Reset	HW Reset
enter_idle_mode	39h	None	Idle Mode Off	Idle Mode Off	Idle Mode Off
enter_invert_mode	21h	None	Display Inversion Off	Display Inversion Off	Display Inversion Off
enter_normal_mode	13h	None	Normal Display mode On	Normal Display mode On	Normal Display mode On
enter_partial_mode	12h	None	Normal Display Mode On	Normal Display Mode On	Normal Display Mode On
enter_sleep_mode	10h	None	Sleep Mode On	Sleep Mode On	Sleep Mode On
exit_idle_mode	38h	None	Idle Mode Off	Idle Mode Off	Idle Mode Off
exit_invert_mode	20h	None	Display Inversion Off	Display Inversion Off	Display Inversion Off
exit_sleep_mode	11h	None	Sleep Mode On	Sleep Mode On	Sleep Mode On
get_3D_control	3Fh	1st and 2nd	00h	00h	00h
get_address_mode	0Bh	1st	Refer to corresponding command parameters	Refer to corresponding command parameters	Refer to corresponding command parameters
get_blue_channel	08h	1st	00h	00h	00h
get_CABC_min_brightness	5Fh	1st and 2nd	00h	00h	00h
get_compression_mode	03h	1st	Refer to corresponding command parameters	Refer to corresponding command parameters	Refer to corresponding command parameters
get_control_display	54h	1st	00h	00h	00h
get_diagnostic_result	0Fh	1st	00h	00h	00h
get_display_brightness	52h	1st and 2nd	00h	00h	00h
get_display_mode	0Dh	1st	Refer to corresponding command parameters	Refer to corresponding command parameters	Refer to corresponding command parameters
get_error_count_on_DSI	05h	1st	00h	00h	00h
get_green_channel	07h	1st	00h	00h	00h
get_image_checksum_ct	15h	2	Normal Display mode On	FFFFh	FFFFh
get_image_checksum_rgb	14h	2	Normal Display mode On	FFFFh	FFFFh

Command	Hex Code	Parameters	Default Modes and Values, Hex		
			Power-on Sequence	SW Reset	HW Reset
get_pixel_format	0Ch	1st	Refer to corresponding command parameters	Refer to corresponding command parameters	Refer to corresponding command parameters
get_power_mode	0Ah	1st	08h	08h	08h
get_power_save	56h	1st	00h	00h	00h
get_red_channel	06h	1st	00h	00h	00h
get_scanline	45h	1st and 2nd	Refer to corresponding command parameters	Refer to corresponding command parameters	Refer to corresponding command parameters
get_signal_mode	0Eh	1st	Refer to corresponding command parameters	Refer to corresponding command parameters	Refer to corresponding command parameters
nop	00h	None	N/A	N/A	N/A
read_DDB_continue	A8h	all	See [MIPI04]		
read_DDB_start	A1h	all	See [MIPI04]		
read_memory_continue	3Eh	all	Random values	Not cleared	Not cleared
read_memory_start	2Eh	all	Random values	Not cleared	Not cleared
read_PPS_continue	A9h	all	PPS Table 1 values, if present. If no table is present, FFh.	PPS Table 1 values, if present. If no table is present, FFh.	PPS Table 1 values, if present. If no table is present, FFh.
read_PPS_start	A2h	all	PPS Table 1 values, if present. If no table is present, FFh.	PPS Table 1 values, if present. If no table is present, FFh.	PPS Table 1 values, if present. If no table is present, FFh.
set_3D_control	3Dh	1st and 2nd	00h	00h	00h
set_address_mode	36h	1st	00000000b	No change from the value before SW reset	00000000b
set_CABC_min_brightness	5Eh	1st and 2nd	00h	00h	00h
set_column_address	2Ah	1st	00h	00h	00h
		2nd	00h	00h	00h
		3rd	The frame memory column address corresponding to the last vertical line.	If set_address_mode's B5 = 0;The frame memory column address corresponding to the last vertical	The frame memory column address corresponding to the last vertical line.

Command	Hex Code	Parameters	Default Modes and Values, Hex		
			Power-on Sequence	SW Reset	HW Reset
		4th		line. If set_address_mode’s B5 = 1; The frame memory column address corresponding to the last horizontal line.	
set_display_brightness	51h	1st and 2nd	00h	00h	00h
set_display_off	28h	None	Display Off	Display Off	Display Off
set_display_on	29h	None	Display Off	Display Off	Display Off
set_gamma_curve	26h	1st	01h	01h	01h
set_page_address	2Bh	1st	00h	00h	00h
		2nd			
		3rd	The frame memory page address corresponding to the last horizontal line.	If set_address_mode’s B5 = 0; The frame memory page address corresponding to the last horizontal line. If set_address_mode’s B5 = 1; The frame memory page address corresponding to the last vertical line.	The frame memory page address corresponding to the last horizontal line.
		4th			
set_partial_columns	31h	1st	00h	00h	00h
		2nd			
		3rd	The frame memory column address corresponding to the last vertical line.	The frame memory column address corresponding to the last vertical line.	The frame memory column address corresponding to the last vertical line.
set_partial_rows	30h	1st	00h	00h	00h
		2nd			
		3rd	The frame memory page address corresponding to the last horizontal line.	The frame memory page address corresponding to the last horizontal line.	The frame memory page address corresponding to the last horizontal line.
		4th			
set_pixel_format	3Ah	1st	07h	07h	07h

Command	Hex Code	Parameters	Default Modes and Values, Hex		
			Power-on Sequence	SW Reset	HW Reset
set_scroll_area	33h	1st	00h	00h	00h
		2nd	00h	00h	00h
		3rd	The frame memory page address corresponding to the last horizontal line.	The frame memory page address corresponding to the last horizontal line.	The frame memory page address corresponding to the last horizontal line.
		4th			
		5th	00h	00h	00h
		6th	00h	00h	00h
set_scroll_start	37h	1st	00h	00h	00h
		2nd	00h	00h	00h
set_tear_off	34h	None	TE line output OFF	TE line output OFF	TE line output OFF
set_tear_on	35h	1st			
set_tear_scanline	44h	1st	00h	00h	00h
		2nd	00h	00h	00h
set_vsync_timing	40h	1st	00h	00h	00h
soft_reset	01h	None	N/A	N/A	N/A
write_control_display	53h	1st	00h	00h	00h
write_LUT	2Dh	all	Random values	Contents of LUT protected	Random values
write_memory_continue	3Ch	all	Random values	Not cleared	Not cleared
write_memory_start	2Ch	all	Random values	Not cleared	Not cleared
write_power_save	55h	1st	00h	00h	00h

5.8 Image Data Compression

This section, including Section 5.8.1, shall apply for displays using Architecture Type 1, Type 2, or Type 3, when DSI interface [MIPI03] forms the link between the host processor and display device.

A command mode display with frame memory may optionally support display stream compression if the decoder is implemented on the display. When the compression scheme is enabled with the Compression Mode Command short packet, defined in [MIPI03], the display shall treat all incoming pixel data as a compressed bitstream. The Compression Mode Command is specified in more detail in [MIPI03].

Figure 1 shows possible data flow paths for when compression mode is set as "enabled" [MIPI03] (Section 6.13). It is an implementation choice if Data Path 1 or Data Path 2 is used by a particular display driver. If no frame memory is present, Data Path 2 is the only choice.

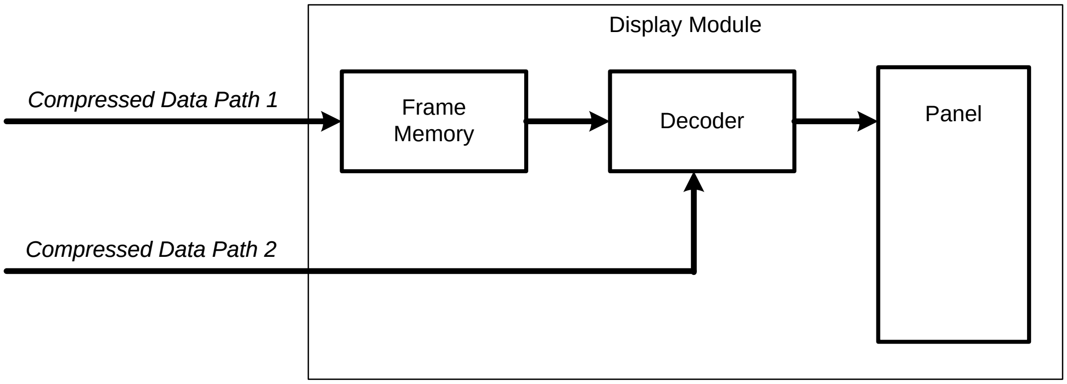


Figure 17 Compressed Data Flow

In Figure 17, displays designed to use Architecture Type 1 or Type 2, with decoder implemented, can receive compressed image data using Path 1 and store it to frame memory. The decoder can then decompress the compressed image data from frame memory to update the panel. A display with frame memory might be able to support switching between Path 1 and Path 2 if video stream is also supported.

In Figure 17, displays designed to use Architecture Type 3, with decoder implemented, can receive compressed image data using Path 2. See [MIPI03] for Compression Mode definition and details about compressed data transport in video mode using Path 2. Switching between modes shall not cause any abnormal behavior or visual defects on the panel.

5.8.1 Display Stream Compression Transport in Command Mode

When compression mode status is set as "enabled" in Command Mode, the compression scheme shall become active and the compressed pixel data shall be transmitted to the display using Long Packet format. See Long Packet and Command mode definition from [MIPI03]. In this case, the first byte of the payload shall be a write_memory_start or a write_memory_continue command and the display shall treat all following image data as compressed data. Data bytes following any other commands in Long Packet type shall not be treated as compressed data.

Figure 18 shows compressed data transportation in protocol level for Architecture Type 1 or Type 2 displays using Command Mode.

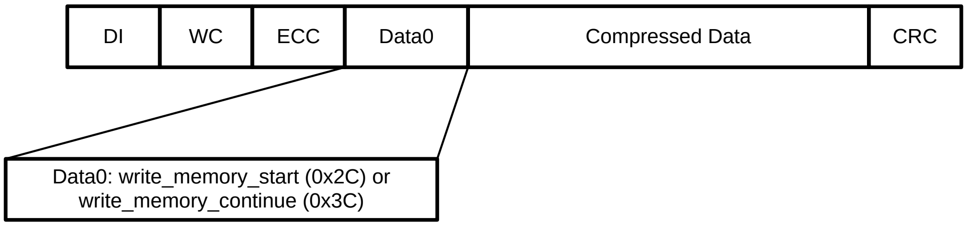

DI
WC
ECC
Data0
Compressed Data
CRC
Data0: write_memory_start (0x2C) or write_memory_continue (0x3C)

Figure 18 Compressed Data Transportation in Command Mode

In Figure 18, Long Packet data type 0x39 is used to carry compressed image data over DSI system [MIPI03].

6 Command Description

This section defines the commands supported by a display module implementing MIPI Alliance Specifications for display interfaces.

All commands consist of a single 8-bit byte, in some cases accompanied by parameters that supply necessary information for the correct execution of the command. Generally, the command and accompanying parameter bytes are transferred using serial or parallel bits 0 through 7 of the display interface, regardless of the physical interface width and architecture. The only exceptions are the read_memory_continue, read_memory_start, write_memory_continue, and write_memory_start commands in a DBI system (see [MIPI02]). The full width of the display interface may be used by these commands. See Section 6.31, Section 6.32, Section 6.57, and Section 6.58 for the command descriptions.

Command flow charts in this section use the symbols defined in Figure 19.

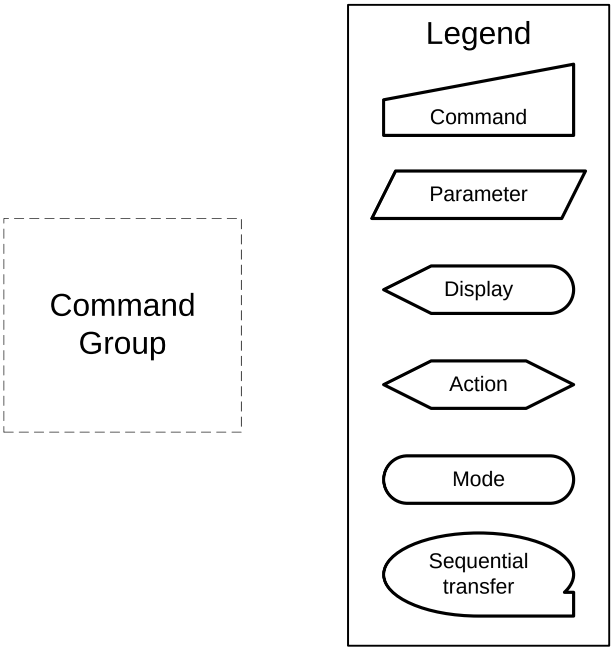


Figure 19 Flowchart Legend

6.1 enter_idle_mode

Interface All

Command 39h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	0	0	1	39h

Description

This command causes the display module to enter Idle Mode.

In Idle Mode, color expression is reduced. Colors are shown on the display device using the MSB of each of the R, G and B color components in the frame memory.

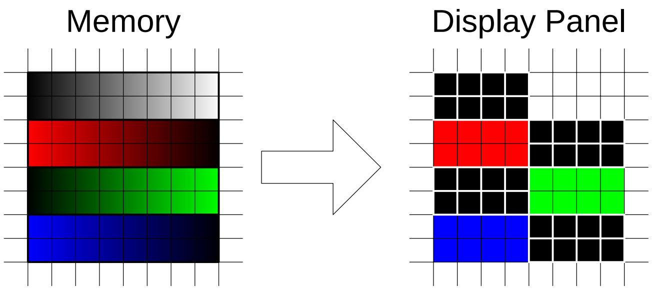


Figure 20 enter_idle_mode Example

Table 5 enter_idle_mode Memory Content vs. Display Color

Color	R_7 R_6 R_5 R_4 R_3 R_2 R_1 R_0	G_7 G_6 G_5 G_4 G_3 G_2 G_1 G_0	B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0
Black	0XXXXXXX	0XXXXXXX	0XXXXXXX
Blue	0XXXXXXX	0XXXXXXX	1XXXXXXX
Red	1XXXXXXX	0XXXXXXX	0XXXXXXX
Magenta	1XXXXXXX	0XXXXXXX	1XXXXXXX
Green	0XXXXXXX	1XXXXXXX	0XXXXXXX
Cyan	0XXXXXXX	1XXXXXXX	1XXXXXXX
Yellow	1XXXXXXX	1XXXXXXX	0XXXXXXX
White	1XXXXXXX	1XXXXXXX	1XXXXXXX

Restrictions

This command has no effect when the display module is already in Idle Mode.

Flow Chart

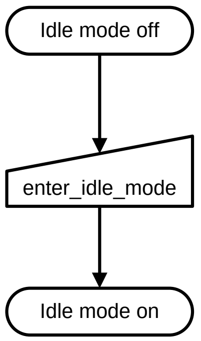


Figure 21 enter_idle_mode Flow Chart

6.2 enter_invert_mode

Interface All

Command 21h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	0	0	0	1	21h

Description

This command causes the display module to invert the image data only on the display device. The frame memory contents remain unchanged. No status bits are changed.

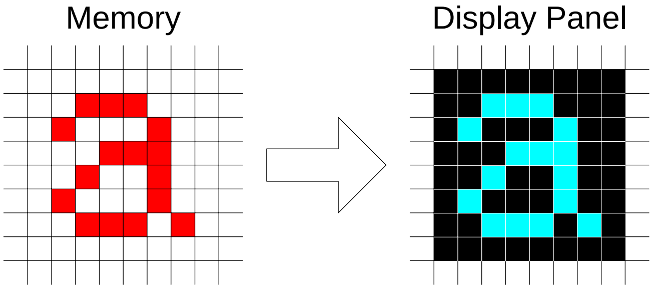


Figure 22 enter_invert_mode Example

Restrictions

This command has no effect when the display module is already inverting the display image.

Flow Chart

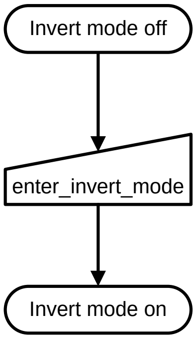


Figure 23 enter_invert_mode Flow Chart

6.3 enter_normal_mode

Interface All

Command 13h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	1	0	0	1	1	13h

Description

This command causes the display module to enter the Normal mode.

Normal Mode is defined as Partial Display mode and Scroll mode are off.

The host processor sends PCLK, HS and VS information to Type 2 display modules two frames before this command is sent when the display module is in Partial Display Mode.

Restrictions

This command has no effect when Normal Display mode is already active.

Flow Chart

See Section 6.44 and Section 6.47 for details of when to use this command.

6.4 enter_partial_mode

Interface All

Command 12h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	1	0	0	1	0	12h

Description

This command causes the display module to enter the Partial Display Mode. The Partial Display Mode window is described by the set_partial_columns and set_partial_rows commands. See Section 6.44 and Section 6.45, respectively, for details. A display module should not implement enter_partial_mode in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

To leave Partial Display Mode, the enter_normal_mode command should be written.

The host processor continues to send PCLK, HS and VS information to a Type 2 display module for two frames after this command is sent when the display module is in Normal Display Mode.

Restrictions

This command has no effect when Partial Display Mode is already active.

Flow Chart

See Section 6.44.

6.5 enter_sleep_mode

Interface	All
Command	10h
Parameters	None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	1	0	0	0	0	10h

Description

This command causes the display module to enter the Sleep mode.

In this mode, all unnecessary blocks inside the display module are disabled except interface communication. This is the lowest power mode the display module supports.

DBI or DSI Command Mode remains operational and the frame memory maintains its contents. The host processor continues to send PCLK, HS and VS information to Type 2 and Type 3 display modules for two frames after this command is sent when the display module is in Normal mode.

Restrictions

This command has no effect when the display module is already in Sleep mode.

The host processor must wait five milliseconds before sending any new commands to a display module following this command to allow time for the supply voltages and clock circuits to stabilize.

The host processor must wait 120 milliseconds after sending an exit_sleep_mode command before sending an enter_sleep_mode command.

Flow Chart

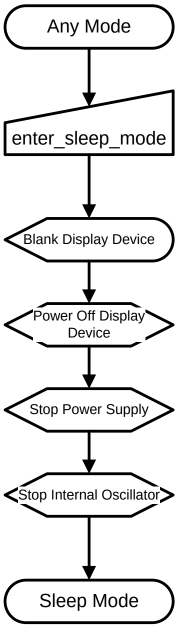


Figure 24 enter_sleep_mode Flow Chart

6.6 exit_idle_mode

Interface All

Command 38h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	0	0	0	38h

Description

This command causes the display module to exit Idle mode.

Restrictions

This command has no effect when the display module is not in Idle mode.

Flow Chart

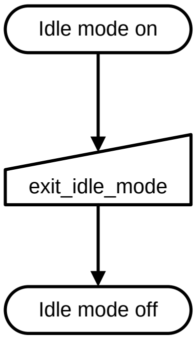


Figure 25 exit_idle_mode Flow Chart

6.7 exit_invert_mode

Interface All

Command 20h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	0	0	0	0	20h

Description

This command causes the display module to stop inverting the image data on the display device. The frame memory contents remain unchanged. No status bits are changed.

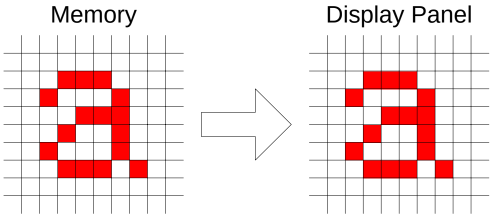


Figure 26 exit_invert_mode Example

Restrictions

This command has no effect when the display module is not inverting the display image.

Flow Chart

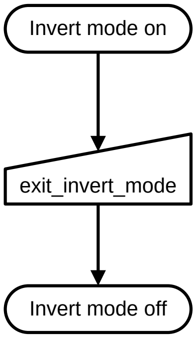


Figure 27 exit_invert_mode Flow Chart

6.8 exit_sleep_mode

Interface All

Command 11h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	1	0	0	0	1	11h

Description

This command causes the display module to exit Sleep mode. All blocks inside the display module are enabled.

The host processor sends PCLK, HS and VS information to Type 2 and Type 3 display modules two frames before this command is sent when the display module is in Normal Mode.

Restrictions

This command shall not cause any visible effect on the display device when the display module is not in Sleep mode.

The host processor must wait five milliseconds after sending this command before sending another command. This delay allows the supply voltages and clock circuits to stabilize.

The host processor must wait 120 milliseconds after sending an exit_sleep_mode command before sending an enter_sleep_mode command.

The display module loads the display module's default values to the registers when exiting the Sleep mode. There shall not be any abnormal visual effect on the display device when loading the registers if the factory default and register values are the same or when the display module is not in Sleep mode.

The display module runs the self-diagnostic functions after this command is received. See Section 5.3 for a description of the self-diagnostic functions.

Flow Chart

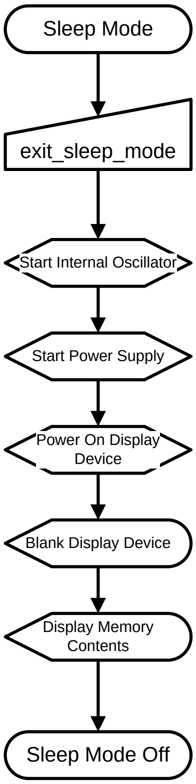


Figure 28 exit_sleep_mode Flow Chart

6.9 get_3D_control

Interface All

Command 3Fh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	1	1	1	3Fh

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	0	0	3DL/R	3DVSYNC	3DFMT[1:0]		3DMODE[1:0]		XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	0	0	0	0	0	0	0	0	00h

Description

Support for get_3D_control is optional. However, if get_3D_control is supported, it shall be implemented as described in this section.

The display module returns the values of the 3D Control Function (see [MIPI05]).

In 3D Mode, certain commands operate differently (see Table 9).

D7 – Reserved, set to ‘0’.

D6 – Reserved, set to ‘0’.

3DL/R – Left / Right Order

‘0’ = Data sent left eye first, right eye next.

‘1’ = Data sent right eye first, left eye next.

3DVSYNC – Second VSYNC Enabled between Left and Right images

‘0’ = No sync pulses between left and right data.

‘1’ = Sync pulse (HSYNC, VSYNC, blanking) between left and right data.

3DFMT[1:0] – Stereoscopic Image Format

‘00’ = Line (alternating lines of left and right data).

‘01’ = Frame (alternating frames of left and right data).

’10’ = Pixel (alternating pixels of left and right data).

’11’ = Reserved

3DMODE[1:0] – 3D Mode On / Off, Display Orientation

‘00’ = 3D Mode Off (2D Mode On).

‘01’ = 3D Mode On, Portrait Orientation.

’10’ = 3D Mode On, Landscape Orientation.

’11’ = Reserved.

Restrictions

None

Flow Chart

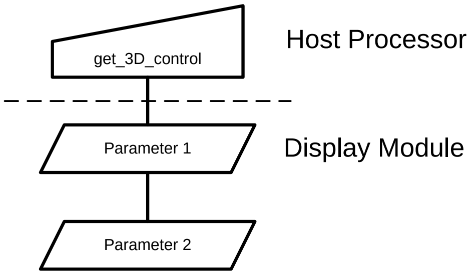


Figure 29 get_3D_control Flow Chart

6.10 get_address_mode

Interface All

Command 0Bh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	1	0	1	1	0Bh

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Description

The display module returns the current status.

In 2D mode, a device shall use the parameter bit definitions for D7 through D0 as provided.

In 3D Mode, a device shall set inapplicable bits to '0'. Which bits are relevant in 3D Mode is implementation-specific. The manufacturer of a device shall describe any such implementation-specific behavior in the product datasheet.

If the device supports compression and [VESA01] is selected as the active compression mode algorithm, functionality of some bits cannot be guaranteed. Thus, all inapplicable bits are set to ‘0’. If the device supports a vendor specific algorithm, the manufacturer of the device shall define which bits shall be supported when compression mode status is ‘enabled’.

D7 – Page Address Order

If [VESA01] is selected for active compression algorithm, this bit is set as ‘0’

‘0’ = Top to Bottom

‘1’ = Bottom to Top

D6 – Column Address Order

If [VESA01] is selected for active compression algorithm, this bit is set as ‘0’

‘0’ = Left to Right

‘1’ = Right to Left

D5 - Page/Column Order

If [VESA01] is selected for active compression algorithm, this bit is set as ‘0’

‘0’ = Normal Mode

‘1’ = Reverse Mode

D4 – Line Address Order

‘0’ = LCD Refresh Top to Bottom

‘1’ = LCD Refresh Bottom to Top

D3 – RGB/BGR Order

‘0’ = RGB

‘1’ = BGR

D2 – Display Data Latch Data Order

‘0’ = LCD Refresh Left to Right

‘1’ = LCD Refresh Right to Left

Not applicable for display modules scanned line by line

D1 – Flip Horizontal

This bit flips the image shown on the display device left to right. No change is made to the frame memory.

‘0’ = Normal

‘1’ = Flipped

D0 – Flip Vertical

This bit flips the image shown on the display device top to bottom. No change is made to the frame memory.

‘0’ = Normal

‘1’ = Flipped

Restrictions

None

Flow Chart

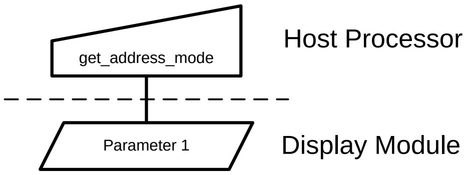


Figure 30 get_address_mode Flow Chart

6.11 get_blue_channel

Interface All

Command 08h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	1	0	0	0	08h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	B7	B6	B5	B4	B3	B2	B1	B0	XXh

Description

The display module returns the blue component value of the first pixel in the active frame. This command is only valid for Type 2 and Type 3 display modules.

In 2D mode, a device shall use the bit definitions for D7 through D0 as provided. B7 is the MSB and B0 is the LSB.

Only the relevant bits are used according to the pixel format; unused bits are set to ‘0’

Examples:

- 12-bit format: B3 is MSB and B0 is LSB. B[7:4] are set to ‘0’.
- 16-bit format: B5 is MSB, B1 is LSB and B7, B6 and B0 are set to ‘0’.
- 18-bit format: B5 is MSB and B0 is LSB. B7 and B6 are set to ‘0’.
- 24-bit format: B7 is MSB and B0 is LSB. All bits are used.

In 3D Mode, get_blue_channel shall return the blue component of the first pixel of the active frame in memory. See Section 6.35 for bit order dependency.

If Compression Mode bit CMODE = 1:

This command returns the first of three eight-bit values.

Restrictions

None

Flow Chart

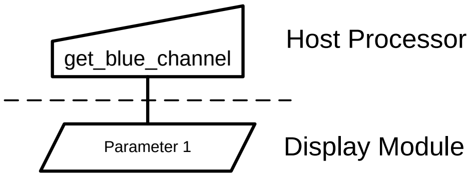


Figure 31 get_blue_channel Flow Chart

6.12 get_CABC_min_brightness

Interface All

Command 5Fh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	1	1	1	1	5Fh

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	CMB15	CMB14	CMB13	CMB12	CMB11	CMB10	CMB9	CMB8	xxh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	CMB7	CMB6	CMB5	CMB4	CMB3	CMB2	CMB1	CMB0	xxh

Description

This command returns the minimum brightness value for the CABC function.

Value 0000h means the lowest brightness level, and value FFFFh means the highest brightness level.

Note: It is up to display manufacturer to determine the implementation of this register and background logic.

Restrictions

This command is only for use with LCDs with CABC implementations. Only one parameter is returned for devices that support 8-bit brightness levels. Two parameters are returned for devices that support between 9-bit and 16-bit brightness levels.

Flow Chart

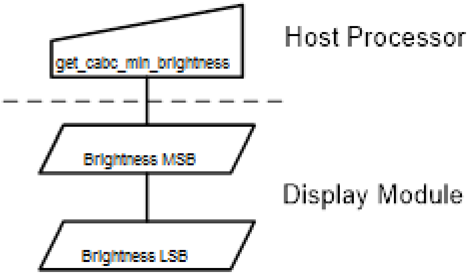


Figure 32 get_CABC_min_brightness Flow Chart

6.13 get_compression_mode

Interface All

Command 03h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	0	1	1	03h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	0	0	PPSSEL[1:0]		0	ALGID[1:0]		CMODE	xxh

Description

Support for get_compression_mode is optional. If compression support is implemented the display module returns the current status of compression mode.

Display shall use parameter bit definitions as provided.

D7 – Reserved, set as ‘0’

D6 – Reserved, set as ‘0’

PPSSEL[1:0] – PPS Table selector

‘00’ = PPS Table 1 (default)

‘01’ = PPS Table 2

‘10’ = PPS Table 3

‘11’ = PPS Table 4

D3 – Reserved, set as ‘0’

ALGID[1:0] – Algorithm identifier

‘00’ = VESA DSC Standard 1.1

‘01’ = Reserved

‘10’ = Reserved

‘11’ = Vendor specific algorithm

CMODE – Compression mode enable/disable

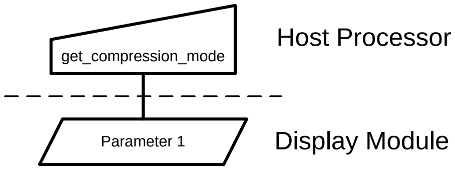
‘0’ = Compression mode is disabled (default)

‘1’ = Compression mode is enabled

Restrictions

None

687 **Flow Chart**



688

689 **Figure 33 get_compression_mode Flow Chart**

6.14 get_control_display

Interface All

Command 54h

Parameters See below

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	0	1	0	0	54h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	x	x	BCTRL	x	DD	BL	x	x	xxh

This command returns the current brightness control mode of the display.

BCTRL – Brightness Control Block On/Off
‘0’ = Off (Brightness register is 0000h)
‘1’ = On (Brightness registers are active)

DD – Display Dimming
‘0’ = Display Dimming Off
‘1’ = Display Dimming On

BL – Backlight On/Off (For LCD)
‘0’ = Off (completely turn off backlight circuit)
‘1’ = On

Bit marked as ‘x’ are reserved for display manufacturer’s own usage.
Note: It is up to display manufacturer to determine about the implementation of this register and background logic.

Restrictions

None

Flow Chart

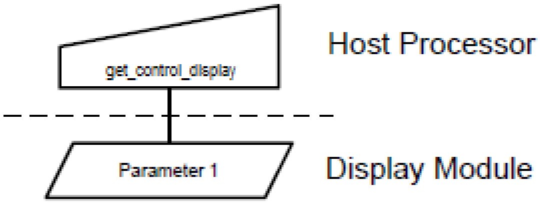


Figure 34 get_control_display Flow Chart

6.15 get_diagnostic_result

Interface All

Command 0Fh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	1	1	1	1	0Fh

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	0	0	0	0	XXh

Description

The display module returns the self-diagnostic results following a Sleep Out command. See Section 5.3 for a description of the status results.

D7 – Register Loading Detection

D6 – Functionality Detection

D5 – Chip Attachment Detection

Set to ‘0’ if feature unimplemented.

D4 – Display Glass Break Detection

Set to ‘0’ if feature unimplemented.

D[3:0] – Reserved

Set to ‘0’.

Restrictions

None

Flow Chart

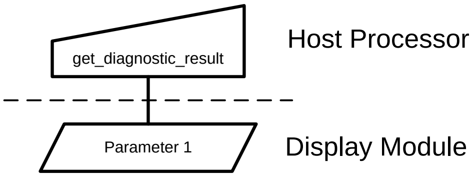


Figure 35 get_diagnostic_result Flow Chart

6.16 get_display_brightness

Interface All

Command 52h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	0	0	1	0	52h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	DBV15	DBV14	DBV13	DBV12	DBV11	DBV10	DBV9	DBV8	xxh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	DBV7	DBV6	DBV5	DBV4	DBV3	DBV2	DBV1	DBV0	xxh

Description

This command returns the current brightness value of the display, set by the set_display_brightness (51h) command.

In addition:

- DBV[15:0] = 0000h when the display is in Sleep In mode.
- DBV[15:0] = 0000h when bit ‘BCTRL’ of write_control_display (53h) is ‘0’.
- DBV[15:0] = the manual set brightness specified with the set_display_brightness (51h) when bit BCTRL of write_control_display (53h) is 1.

Value 0000h means the lowest brightness level, and value FFFFh means the highest brightness level.

Note: It is up to display manufacturer to determine the implementation of this register and background logic.

Restrictions

Only one parameter is returned for devices that support 8-bit brightness levels. Two parameters are returned for devices that support between 9-bit and 16-bit brightness levels.

Flow Chart

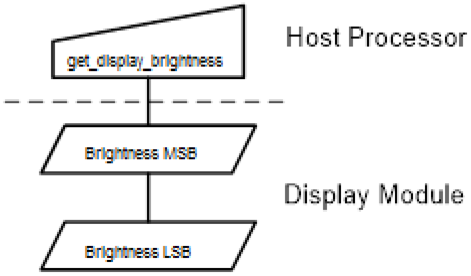


Figure 36 get_display_brightness Flow Chart

6.17 get_display_mode

Interface All

Command 0Dh

Parameters See the following description.

Command

									Hex
Direction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Code**
H→D	0	0	0	0	1	1	0	1	0Dh

Parameter

									Hex
Direction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Code**
D→H	D7	0	D5	0	0	D2	D1	D0	XXh

Description

The display module returns the Display Image Mode status.

D7 – Vertical Scrolling Status

‘0’ = Vertical Scrolling is Off.

‘1’ = Vertical Scrolling is On.

D6 – Reserved

Set to ‘0’.

D5 – Inversion On/Off

‘0’ = Inversion is Off.

‘1’ = Inversion is On.

D4 – Reserved

Set to ‘0’.

D3 – Reserved

Set to ‘0’.

D[2:0] – Gamma Curve Selection

Table 6 Gamma Curve Selection

Gamma Curve Selection	D2	D1	D0	Gamma Set (26h) Parameter
Gamma Curve 1	0	0	0	GC0
Gamma Curve 2	0	0	1	GC1
Gamma Curve 3	0	1	0	GC2
Gamma Curve 4	0	1	1	GC3
Reserved	1	0	0	Reserved
Reserved	1	0	1	Reserved
Reserved	1	1	0	Reserved

Gamma Curve Selection	D2	D1	D0	Gamma Set (26h) Parameter
Reserved	1	1	1	Reserved

Restrictions

None

Flow Chart

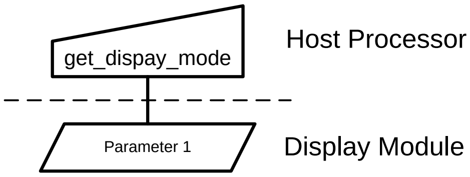


Figure 37 get_display_mode Flow Chart

6.18 get_error_count_on_DSI

Interface DSI

Command 05h

Parameters See below

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	1	0	1	05h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	P7	P6	P5	P4	P3	P2	P1	P0	xxh

This command returns the number of corrupted packets previously received on the DSI link. When multiple DSI links are connected to one Display Driver or Display Controller, this command returns one parameter for each DSI link, where parameter 1 contains the error count for DSI link 0, parameter 2 contains the error count for DSI link 1, etc.

For each parameter, P[7] is set to ‘1’ if there is overflow with P[6..0] bits.

For each parameter, P[7..0] bits are set to ‘0’s (as well as get_signal_mode (0Eh)’s D0 is set to ‘0’ at the same time) after the parameter information is sent, indicating that the read function is completed.

Restrictions

Only ECC single-bit errors, ECC multi-bit errors and Checksum errors [MIPI03] are included in the error counter functionality.

Flow Chart

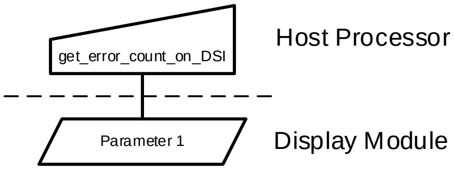


Figure 38 get_error_count_on_DSI Flow Chart

6.19 get_green_channel

Interface All

Command 07h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	1	1	1	07h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	G7	G6	G5	G4	G3	G2	G1	G0	XXh

Description

The display module returns the green component value of the first pixel in the active frame. This command is only valid for Type 2 and Type 3 display modules.

In 2D mode, a device shall use the bit definitions for D7 through D0 as provided. G7 is the MSB and G0 is the LSB.

Only the relevant bits are used according to the pixel format; unused bits are set to ‘0’

Examples:

- 12-bit format: G3 is MSB and G0 is LSB. G[7:4] are set to ‘0’.
- 16-bit format: G5 is MSB, G0 is LSB and G7 and G6 are set to ‘0’.
- 18-bit format: G5 is MSB and G0 is LSB. G7 and G6 are set to ‘0’.
- 24-bit format: G7 is MSB and G0 is LSB. All bits are used.

In 3D Mode, get_green_channel shall return the green component of the first pixel of the active frame in memory. See Section 6.35 for bit order dependency.

If Compression Mode bit CMODE = 1:

This command returns the second of three eight-bit values.

Restrictions

None

Flow Chart

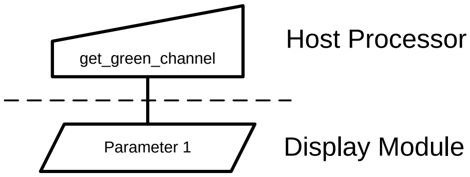


Figure 39 get_green_channel Flow Chart

6.20 get_image_checksum_ct

Interface All

Command 15h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	1	0	1	0	1	15h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	N15	N14	N13	N12	N11	N10	N9	N8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	N7	N6	N5	N4	N3	N2	N1	N0	XXh

Description

The display module returns, N, the current value of a checksum register containing the output of a checksum calculation located in a display driver IC and specified in [MIPI03].

This mode is intended for test purposes to verify if a bitstream has been decompressed and reconstructed to a value defined by the display stream coding system. However a panel module shall return a value that is unspecified in normal display operation in a system device.

In Sleep Mode, the value returned by get_image_checksum_ct is undefined.

In 2D mode, the checksum is a result of the calculation on one frame of data.

In 3D mode, the checksum can support any pixel ordering or frame-sequential data ordering. For all non-frame sequential stereoscopic formats (for example, 3DFMT not equal to 01h) the checksum of the frame includes pixel data from the left eye and the right eye. For 3D operation with Frame-sequential pixel formats, the checksum includes pixel data from only the left eye or the right eye. Refer to [MIPI05] for non-temporal mode pixel ordering and for temporal mode frame ordering between the left eye and right eye.

Restrictions

None

Flow Chart

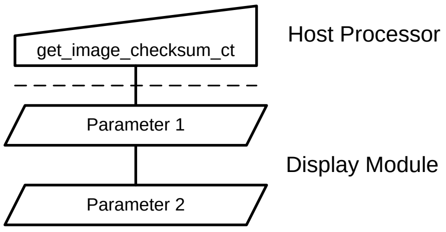


Figure 40 get_image_checksum_ct Flow Chart

6.21 get_image_checksum_rgb

Interface All

Command 14h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	1	0	1	0	0	14h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	N15	N14	N13	N12	N11	N10	N9	N8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	N7	N6	N5	N4	N3	N2	N1	N0	XXh

Description

The display module returns, N, the current value of a checksum register containing the output of a checksum calculation located in a display driver IC and specified in [MIPI03]

This mode is intended for test purposes to verify if a bitstream has been decompressed and reconstructed to a value defined by the display stream coding system. However a panel module shall return a value that is unspecified in normal display operation in a system device.

In Sleep Mode, the value returned by get_image_checksum_rgb is undefined.

In 2D mode, the checksum is a result of the calculation on one frame of data.

In 3D mode, the checksum can support any pixel ordering or frame-sequential data ordering. For all non-temporal stereoscopic formats the checksum of the frame includes pixel data from the left eye and the right eye. For 3D operation with Frame-sequential pixel formats, the checksum includes pixel data from only the left eye or the right eye. Refer to [MIPI05] for non-temporal mode pixel ordering and for temporal mode frame ordering between the left eye and right eye.

Restrictions

None

Flow Chart

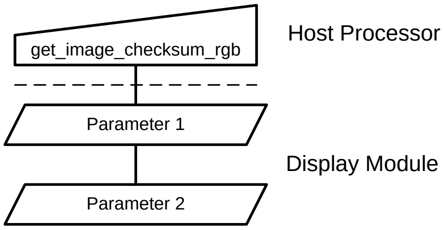


Figure 41 get_image_checksum_rgb Flow Chart

6.22 get_pixel_format

Interface All

Command 0Ch

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	1	1	0	0	0Ch

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	0	D6	D5	D4	0	D2	D1	D0	XXh

Description

This command gets the pixel format for the RGB image data used by the interface.

D[6:4] – DPI Pixel Format Definition

D[2:0] – DBI Pixel Format Definition

D7 and D3 are not used.

The pixel formats are shown in Table 7.

Table 7 Interface Pixel Formats

Pixel Format	D6/D2	D5/D1	D4/D0
Reserved	0	0	0
3 bits/pixel	0	0	1
8 bits/pixel	0	1	0
12 bits/pixel	0	1	1
Reserved	1	0	0
16 bits/pixel	1	0	1
18 bits/pixel	1	1	0
24 bits/pixel	1	1	1

If a particular interface, either DBI or DPI, is not used then the corresponding bits in the parameter returned from the display module are undefined. Therefore, for a DBI display module, the Host shall ignore D[6:4] and for a DPI display module, the Host shall ignore D[2:0].

If Compression Mode bit CMODE = 1:

This feature is not supported, return Reserved.

Restrictions

None

Flow Chart

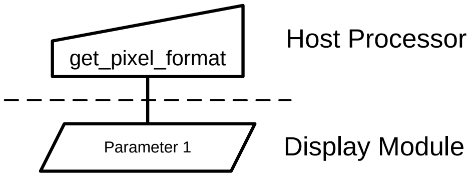


Figure 42 get_pixel_format Flow Chart

6.23 get_power_mode

Interface All

Command 0Ah

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	1	0	1	0	0Ah

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	0	0	XXh

Description

The display module returns the current power mode.

D7 – Reserved

Set to ‘0’

D6 - Idle Mode On/Off

‘0’ = Idle Mode Off.

‘1’ = Idle Mode On.

D5 – Partial Mode On/Off

‘0’ = Partial Mode Off.

‘1’ = Partial Mode On.

D4 – Sleep Mode

‘0’ = Sleep Mode On.

‘1’ = Sleep Mode Off.

D3 – Display Normal Mode On/Off

‘0’ = Display Normal Mode Off.

‘1’ = Display Normal Mode On.

D2 – Display On/Off

‘0’ = Display is Off.

‘1’ = Display is On.

D1 – Reserved

Set to ‘0’

D0 – Reserved

Set to ‘0’

Restrictions

None

Flow Chart

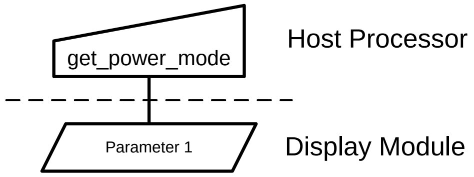


Figure 43 get_power_mode Flow Chart

6.24 get_power_save

Interface All

Command 56h

Parameters See below

Command

									Hex
Direction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Code**
H→D	0	1	0	1	0	1	1	0	56h

Parameter 1

									Hex
Direction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Code**
D→H	x	x	x	x	x	PS2	PS1	PS0	xxh

This command returns the current power save mode of the display. These power saving modes are described in Table 8.

Table 8 Power Saving Modes

PS[2]	PS[1]	PS[0]	Function
0	0	0	Power Save Off
0	0	1	Power Save level: Low
0	1	0	Power Save level: Medium
0	1	1	Power Save: High
1	0	0	Outdoor mode

Bits marked as ‘x’ are reserved for display manufacturer’s own usage.

Note:

> *It is up to the display manufacturer to determine the implementation of this register and background logic.*

Restrictions

None

Flow Chart

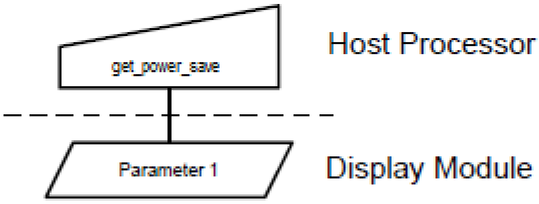


Figure 44 get_power_save Flow Chart

6.25 get_red_channel

Interface All

Command 06h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	1	1	0	06h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	R7	R6	R5	R4	R3	R2	R1	R0	XXh

Description

The display module returns the red component value of the first pixel in the active frame. This command is only valid for Type 2 and Type 3 display modules.

In 2D mode, a device shall use the bit definitions for D7 through D0 as provided. R7 is the MSB and R0 is the LSB.

Only the relevant bits are used according to the pixel format; unused bits are set to ‘0’

Examples:

- 12-bit format: R3 is MSB and R0 is LSB. R[7:4] are set to ‘0’.
- 16-bit format: R5 is MSB, R1 is LSB and R7, R6 and R0 are set to ‘0’.
- 18-bit format: R5 is MSB and R0 is LSB. R7 and R6 are set to ‘0’.
- 24-bit format: R7 is MSB and R0 is LSB. All bits are used.

In 3D Mode, get_red_channel shall return the red component of the first pixel of the active frame in memory. See Section 6.35 for bit order dependency.

If Compression Mode bit CMODE = 1:

This command returns the third of three eight-bit values.

Restrictions

None

Flow Chart

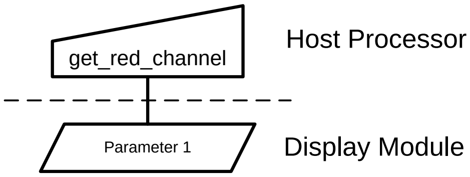


Figure 45 get_red_channel Flow Chart

6.26 get_scanline

Interface All

Command 45h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	0	0	1	0	1	45h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	N15	N14	N13	N12	N11	N10	N9	N8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	N7	N6	N5	N4	N3	N2	N1	N0	XXh

Description

The display module returns the current scanline, N, used to update the display device. The total number of scanlines on a display device is defined as VSYNC + VBP + VACT + VFP. The first scanline is defined as the first line of V Sync and is denoted as Line 0.

In Sleep Mode, the value returned by get_scanline is undefined.

See [MIPI01] for definitions of VSYNC, VBP, VACT, and VFP.

In 2D mode, the scanline value of the display memory and the display panel is the same.

In 3D Mode, the scanline value of the display memory and the display panel can be different; get_scanline shall return the current scanline of the display panel.

Restrictions

None

Flow Chart

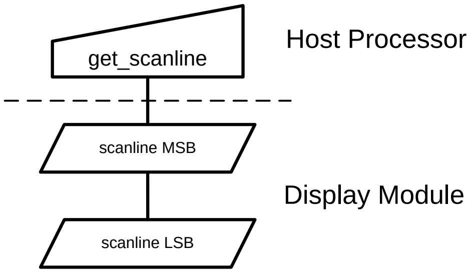


Figure 46 get_scanline Flow Chart

6.27 get_signal_mode

Interface All

Command 0Eh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	1	1	1	0	0Eh

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	0	0	0	0	0	0	X0h

Description

The display module returns the Display Signal Mode.

D7 – Tearing Effect Line

‘0’ = Tearing Effect Line Off.

‘1’ = Tearing Effect On.

D6 – Tearing Effect Line Output Mode.

See [MIPI02] and Section 6.50 for mode definitions.

‘0’ = Mode 0.

‘1’ = Mode 1.

D[5:1] – Reserved

Set to ‘0’.

D0 – Error on DSI.

See [MIPI02] and Section 6.50 for mode definitions.

‘0’ = No error.

‘1’ = Error.

Restrictions

None

Flow Chart

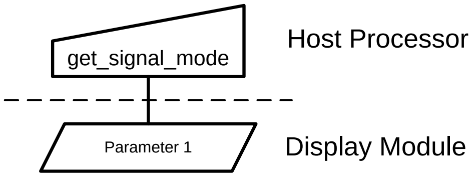


Figure 47 get_signal_mode Flow Chart

6.28 nop

Interface All

Command 00h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	0	0	0	00h

Description

This command does not have any effect on the display module. The nop command may be used to terminate a Frame Memory Read or Frame Memory Write as described in Section 6.31 and Section 6.57, respectively.

Restrictions

None

Flow Chart

None

6.29 read_DDB_continue

Interface All

Command A8h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	1	0	1	0	1	0	0	0	A8h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

.

.

.

Parameter N

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Description

See Section 6.30.

Restrictions

A read_DDB_start command should be executed at least once before a read_DDB_continue command to define the read location. Otherwise, data read with a read_DDB_continue command is undefined.

Flow Chart

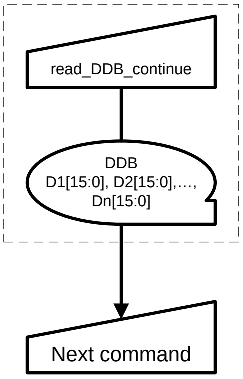


Figure 48 read_DDB_continue Flow Chart

6.30 read_DDB_start

Interface All

Command A1h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	1	0	1	0	0	0	0	1	A1h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D15	D14	D13	D12	D11	D10	D9	D8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Parameter 3

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D15	D14	D13	D12	D11	D10	D9	D8	XXh

Parameter 4

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Parameter 5

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Description

This command reads identifying and descriptive information from the peripheral. This information is organized in the Device Descriptor Block (DDB) stored on the peripheral. The response to this command returns a sequence of bytes that may be any length up to 64K bytes. Note that the returned sequence of bytes does not necessarily correspond to the entire DDB; it may be a portion of a larger block of data.

The format of returned data is as follows:

Parameter 1: MS (most significant) byte of Supplier ID. Supplier ID is a unique value assigned to each peripheral supplier by the MIPI Alliance.

Parameter 2: LS (least significant) byte of Supplier ID.

Parameter 3: MS (most significant) byte of Supplier Elective Data. This is a byte of information that is determined by the supplier. It could include model number or revision information, for example.

Parameter 4: LS (least significant) byte of Supplier Elective Data

Parameter 5: single-byte *Escape or Exit Code* (EEC). The code is interpreted as follows:

- FFh - Exit code – there is no more data in the Descriptor Block
- 00h - Escape code – there is supplier-proprietary data in the Descriptor Block (does not conform to any MIPI Alliance specification)
- Any other value – there is DDB data in the Descriptor Block.

DDBs may contain many more data fields providing information about the peripheral.

In a DSI system, read activity takes the form of two separate transactions across the bus: first the read command read_DDB_start from host processor to peripheral, which includes the bus turn-around token. The peripheral then takes control of the bus and returns the requested data. The peripheral response to read_DDB_start is a Long Packet type, so its length may be up to 64K bytes unless limited by a previous set_max_return_size command.

The response to a read_DDB_start command always starts at the beginning of the Device Descriptor Block. After receiving the first packet and processing the returned DDB data, the host processor may initiate a read_DDB_continue command to access the next portion of the DDB. A read_DDB_continue command begins the next read at the location following the last byte of the previous data read from the DDB.

Subsequent read_DDB_continue commands can be used to read a DDB or supplier-proprietary block of arbitrary size. There is, however, no obligation to read the entire block. The host processor may choose to stop reading after completion of any read_DDB_xxx command.

Restrictions

None

Flow Chart

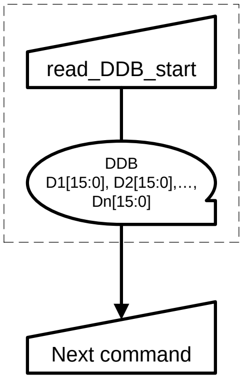


Figure 49 read_DDB_start Flow Chart

6.31 read_memory_continue

Interface All

Command 3Eh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	1	1	0	3Eh

Pixel Data 1

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
D→H	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	P7	P6	P5	P4	P3	P2	P1	P0	XXh

.

.

.

Pixel Data N

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
D→H	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	P7	P6	P5	P4	P3	P2	P1	P0	XXh

Description

This command transfers image data from the display module’s frame memory to the host processor continuing from the location following the previous read_memory_continue or read_memory_start command.

If set_address_mode B5 = 0:

Pixels are read continuing from the pixel location after the read range of the previous read_memory_start or read_memory_continue. The column register is then incremented and pixels are read from the frame memory until the column register equals the End Column (EC) value. The column register is then reset to SC and the page register is incremented. Pixels are read from the frame memory until the page register equals the End Page (EP) value and the column register equals the EC value, or the host processor sends another command.

If set_address_mode B5 = 1:

Pixels are read continuing from the pixel location after the read range of the previous read_memory_start or read_memory_continue. The page register is then incremented and pixels are read from the frame memory until the page register equals the End Page (EP) value. The page register is then reset to SP and the column register is incremented. Pixels are read from the frame memory until the column register equals the End Column (EC) value and the page register equals the EP value, or the host processor sends another command.

If Compression Mode bit CMODE = 1:

> Pixel format of the returned data format might not follow color encoding (defined in Annex A) since image data stored in frame memory is compressed.

See Section 6.38 for descriptions of the Start Column and End Column values.

See Section 6.43 for descriptions of the Start Page and End Page values.

See [MIPI01] and [MIPI02] for color encoding for 8 or 9 bit image data.

Note:

> *The command description shows 16-bit pixel data transferred over a 16-bit bus. Other possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.*

In 3D Mode, read_memory_continue shall return data in the same format that is set by set_3D_control, defining pixel order, and transmission format.

Restrictions

Regardless of the interface format chosen with the set_pixel_format command, the pixel format of the returned data is always the maximum pixel depth supported by the display module. The display module documentation shall describe the maximum pixel depth as well as the format of the data returned by the display module when using this command.

A read_memory_start should follow a set_column_address, set_page_address or set_address_mode to define the read location. Otherwise, data read with read_memory_continue is undefined.

Flow Chart

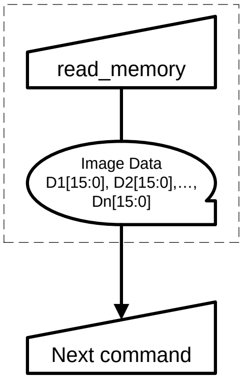


Figure 50 read_memory_continue Flow Chart

6.32 read_memory_start

Interface All

Command 2Eh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	1	1	0	2Eh

Pixel Data 1

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
D→H	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	P7	P6	P5	P4	P3	P2	P1	P0	XXh

.

.

.

Pixel Data N

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
D→H	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	P7	P6	P5	P4	P3	P2	P1	P0	XXh

Description

This command transfers image data from the display module's frame memory to the host processor starting at the pixel location specified by preceding set_column_address and set_page_address commands.

If set_address_mode B5 = 0:

The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixels are read from frame memory at (SC, SP). The column register is then incremented and pixels read from the frame memory until the column register equals the End Column (EC) value. The column register is then reset to SC and the page register is incremented. Pixels are read from the frame memory until the page register equals the End Page (EP) value and the column register equals the EC value, or the host processor sends another command.

If set_address_mode B5 = 1:

The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixels are read from frame memory at (SC, SP). The page register is then incremented and pixels read from the frame memory until the page register equals the End Page (EP) value. The page register is then reset to SP and the column register is incremented. Pixels are read from the frame memory until the column register equals the End Column (EC) value and the page register equals the EP value, or the host processor sends another command.

If Compression Mode bit CMODE = 1:

> Pixel format of the returned data format might not follow color encoding (defined in Annex A) since image data stored in frame memory is compressed.

See Section 6.38 for descriptions of the Start Column and End Column values.

See Section 6.43 for descriptions of the Start Page and End Page values.

See [MIPI01] and [MIPI02] for color encoding for 8 or 9 bit image data.

Note:

> *The command description shows 16-bit pixel data transferred over a 16-bit bus. Other possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.*

In 3D Mode, read_memory_start shall return data in the same format that is set by set_3D_control, defining pixel order, and transmission format.

Restrictions

Regardless of the interface format chosen with the set_pixel_format command, the pixel format of the returned data is always the maximum pixel depth supported by the display module. The display module documentation shall describe the maximum pixel depth as well as the format of the data returned by the display module when using this command.

Flow Chart

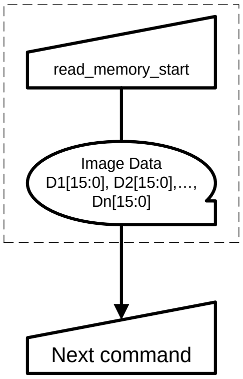


Figure 51 read_memory_start Flow Chart

6.33 read_PPS_continue

Interface All

Command A9h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	1	0	1	0	1	0	0	1	A8h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

.

.

.

Parameter N

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Description

See Section 6.34.

Restrictions

A read_PPS_start command should be executed at least once before a read_PPS_continue command, in order to define the read location. Otherwise, the behavior of data read with a read_PPS_continue command is undefined.

Flow Chart

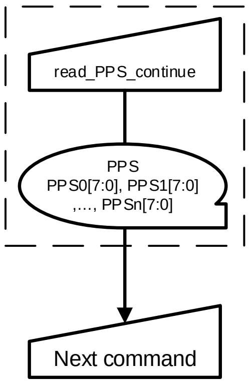


Figure 52 read_PPS_continue Flow Chart

6.34 read_PPS_start

Interface All

Command A2h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	1	0	1	0	0	0	1	0	A2h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D15	D14	D13	D12	D11	D10	D9	D8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Parameter N

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
D→H	D7	D6	D5	D4	D3	D2	D1	D0	XXh

Description

This command initiates the reading of a Picture Parameter Set (PPS) from the peripheral. The response to this command returns a sequence of bytes that may be any length, up to 128 bytes. Note that the returned sequence of bytes does not necessarily correspond to the entire PPS; it may be a portion of a larger block of data.

The format of returned data is as follows:

Parameter 1 through N: the first N bytes of the peripheral's PPS

Parameter N: single-byte *Escape or Exit Code* (EEC). The code is interpreted as follows:

- FFh: Exit Code – There is no more data in the PPS
- Any other value – There is PPS data in the PPS.

Parameters 2 through N: The first N-1 bytes of the peripheral's PPS

A PPS contains parameter values that configure the peripheral's decompression block.

In a DSI system, read activity takes the form of two separate transactions across the bus. First is the read command read_PPS_start from host processor to peripheral, which includes the bus turn-around token. The peripheral then takes control of the bus and returns the requested data. The peripheral response to read_PPS_start is a Long Packet type, so its length may be up to 128 bytes (unless limited by a previous set_max_return_size command).

The response to a read_PPS_start command always starts at the beginning of the PPS. After receiving the first packet and processing the returned PPS data, the host processor may initiate a read_PPS_continue command to access the next portion of the PPS. A read_PPS_continue command begins the next read at the location following the last byte of the previous data read from the PPS.

Subsequent read_PPS_continue commands can be used to read a PPS or supplier-proprietary block of arbitrary size. There is, however, no obligation to read the entire block. The host processor may choose to stop reading after completion of any read_PPS_xxx command.

Restrictions

None

Flow Chart

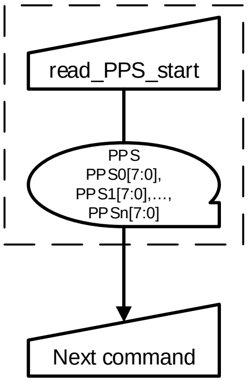


Figure 53 read_PPS_start Flow Chart

6.35 set_3D_control

Interface All

Command 3Dh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	1	0	1	3Dh

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	3DL/R	3DVSYNC	3DFMT[1:0]		3DMODE[1:0]		XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	0	0	0	00h

Description

Support for set_3D_control is optional. However, if set_3D_control is supported, it shall be implemented as described in this section.

The display module sets the values of the 3D Control Function (see [MIPI05]).

In 3D Mode, certain commands operate differently (see Table 9).

D7 – Reserved, set to ‘0’.

D6 – Reserved, set to ‘0’.

3DL/R – Left / Right Order

‘0’ = Data sent left eye first, right eye next.

‘1’ = Data sent right eye first, left eye next.

3DVSYNC – Second VSYNC Enabled between Left and Right images

‘0’ = No sync pulses between left and right data.

‘1’ = Sync pulse (HSYNC, VSYNC, blanking) between left and right data.

3DFMT[1:0] – Stereoscopic Image Format

‘00’ = Line (alternating lines of left and right data).

‘01’ = Frame (alternating frames of left and right data).

’10’ = Pixel (alternating pixels of left and right data).

’11’ = Reserved

3DMODE[1:0] – 3D Mode On / Off, Display Orientation

‘00’ = 3D Mode Off (2D Mode On).

‘01’ = 3D Mode On, Portrait Orientation.

’10’ = 3D Mode On, Landscape Orientation.

’11’ = Reserved.

Table 9 summarizes the commands affected by set_3D_control.

Table 9 DCS 3D Commands

Command	Description
get_address_mode	In 3D Mode, bits not applicable to 3D Mode are set to ‘0’.
get_blue_channel	In 3D Mode, returns the blue component of the first pixel of the active frame in memory.
get_green_channel	In 3D Mode, returns the green component of the first pixel of the active frame in memory.
enter_idle_mode	In 3D Mode, a checksum is available but the value is dependent on pixel ordering in all non-temporal stereoscopic formats, and left-eye or right-eye ordering when using the frame sequential format.
enter_idle_mode	In 3D Mode, a checksum is available but the value is dependent on pixel ordering in all non-temporal stereoscopic formats, and left-eye or right-eye ordering when using the frame sequential format.
get_red_channel	In 3D Mode, returns the red component of the first pixel of the active frame in memory.
get_scanline	In 3D Mode, returns the current scanline of the display panel.
read_memory_start	In 3D Mode, returns data in the same format configured by set_3D_control.
read_memory_continue	In 3D Mode, returns data in the same format configured by set_3D_control.
write_memory_start	In 3D Mode, the pixel order and transmission format are set by set_3D_control.
write_memory_continue	In 3D Mode, the pixel order and transmission format are set by set_3D_control.
set_column_address	In 3D Mode, bits not applicable to 3D Mode are set to ‘0’.
set_page_address	The behavior of this command in 3D Mode, if supported, is specified in the product datasheet.
set_partial_columns	The behavior of this command in 3D Mode, if supported, is specified in the product datasheet.
set_tear_scanline	The behavior of this command in 3D Mode, if supported, is specified in the product datasheet.
set_tear_on	In 3D Mode, this command is affected by 3DVSYNC in set_3D_control.

Restrictions

None

Flow Chart

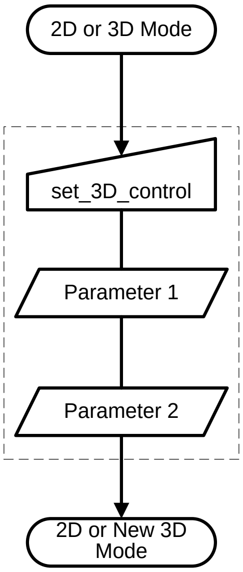


Figure 54 set_3D_control Flow Chart

6.36 set_address_mode

Interface All

Command 36h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	1	1	0	36h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	B7	B6	B5	B4	B3	B2	B1	B0	XXh

Description

This command sets the data order for transfers from the host processor to display module’s frame memory, bits B[7:5], and from the display module’s frame memory to the display device, bits B[4:0].

In 2D mode, a device shall use the parameter bit definitions for B7 through B0 as provided.

In 3D Mode, a device shall set inapplicable bits to '0'. Which bits are relevant in 3D Mode is implementation-specific. The manufacturer of a device shall describe any such implementation-specific behavior in the product datasheet.

If the device supports compression and [VESA01] is selected as the active compression mode algorithm, functionality of some bits cannot be guaranteed. Thus, all inapplicable bits are set to ‘0’. If the device supports a vendor specific algorithm, the manufacturer of the device shall define which bits shall be supported when compression mode status is ‘enabled’.

All bits are valid for peripherals based on the Type 2 display architecture operating in Command Mode, or for peripherals based on the Type 1 display architecture. Bits B5, B4, B2, B1 and B0 have no effect on peripherals based on the Type 2 display architecture operating in Video Mode, or for peripherals based on the Type 3 display architecture.

No status bits are changed.

B7 – Page Address Order

This bit controls the order that Pages of data are transferred from the host processor to the peripheral’s frame memory for a Type 1 or a Type 2 display architecture operating in Command Mode. This bit also controls the Host processor to display device data latching order for a Type 2 or Type 3 display architecture operating in Video Mode.

If [VESA01] is selected for active compression algorithm, this bit is set as ‘0’.

‘0’ = Top to Bottom, Pages transferred from SP to EP

‘1’ = Bottom to Top, Pages transferred from EP to SP

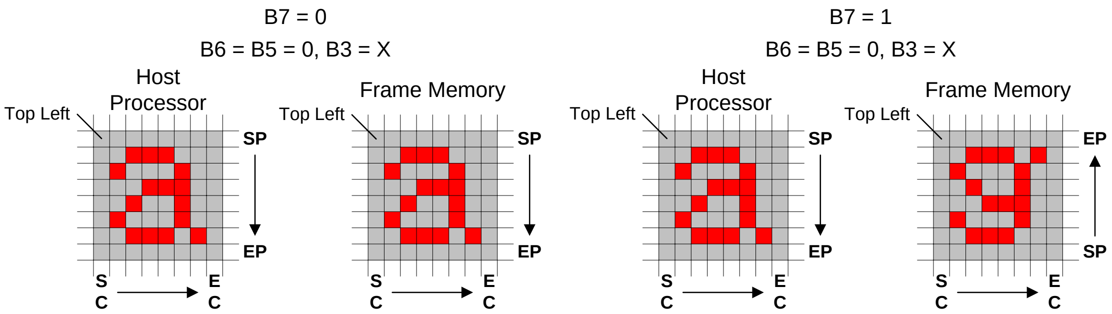


Figure 55 B7 Page Address Order

B6 – Column Address Order

This bit controls the order that Columns of data are transferred from the host processor to the peripheral's frame memory for a Type 1 or Type 2 display architecture operating in Command Mode. This bit also controls the Host processor to display device data latching order for a Type 2 or Type 3 display architecture operating in Video Mode.

If [VESA01] is selected for active compression algorithm, this bit is set as '0'

'0' = Left to Right, Columns transferred from SC to EC

'1' = Right to Left, Columns transferred from EC to SC

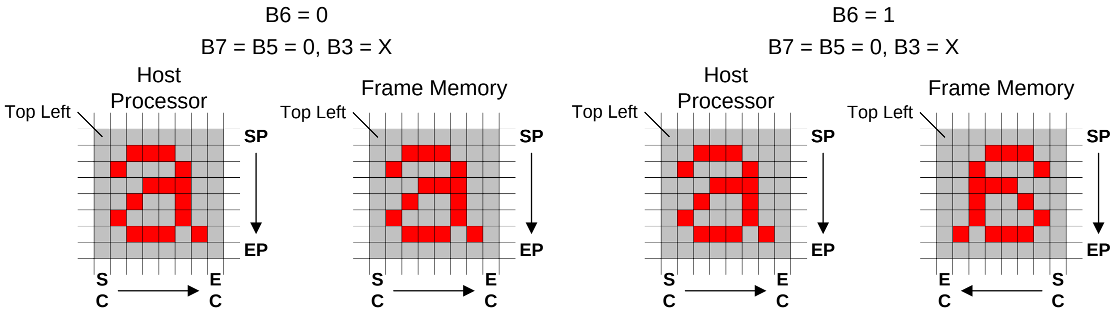


Figure 56 B6 Column Address Order

B5 – Page/Column Addressing Order

This bit controls the order that Columns of data are transferred from the host processor to the peripheral's frame memory.

If [VESA01] is selected for active compression algorithm, this bit is set as '0'

'0' = Normal Mode

See Section 6.58 (B5 = 0) for a description of Normal Mode operation.

'1' = Reverse Mode

See Section 6.58 (B5 = 1) for a description of Reverse Mode operation.

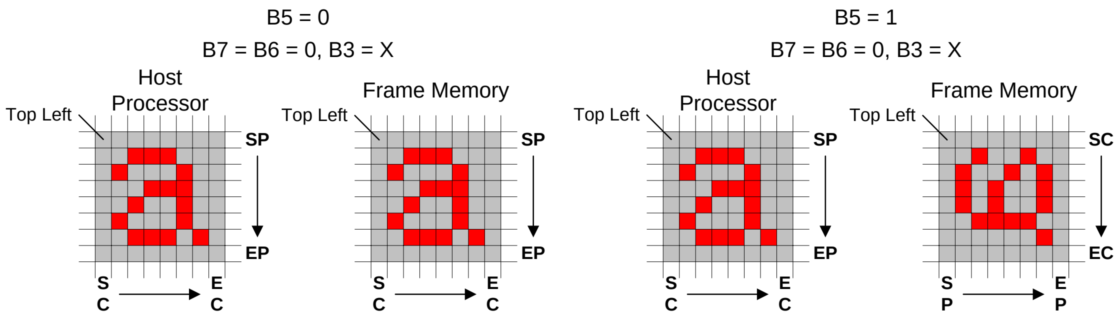


Figure 57 B5 Page/Column Addressing Order

B4 – Display Device Line Refresh Order

This bit controls the display device's horizontal line refresh order. The image shown on the display device is unaffected, regardless of the bit setting.

'0' = Display device is refreshed from the top line to the bottom line

'1' = Display device is refreshed from the bottom line to the top line

B3 – RGB/BGR Order

This bit controls the RGB data latching order transferred from the peripheral's frame memory to the display device for a Type 1 or a Type 2 display architecture operating in Command Mode. This bit also controls the RGB data latching order transfer from the Host processor to the display device for a Type 2 or a Type 3 display architecture operating in Video Mode.

'0' = Pixels sent in RGB order

'1' = Pixels sent in BGR order

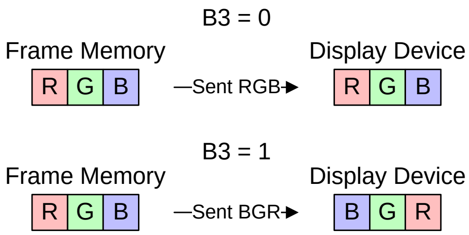


Figure 58 B3 RGB Order

B2 – Display Data Latch Data Order

This bit controls the display device's vertical line data latch order. The image shown on the display device is unaffected, regardless of the bit setting.

'0' = Display device is refreshed from the left side to the right side

'1' = Display device is refreshed from the right side to the left side

Note:

This bit has no visual effect if the display device is refreshed line by line.

B1 – Flip Horizontal

This bit flips the image shown on the display device left to right. No change is made to the frame memory.

‘0’ = Normal

‘1’ = Flipped

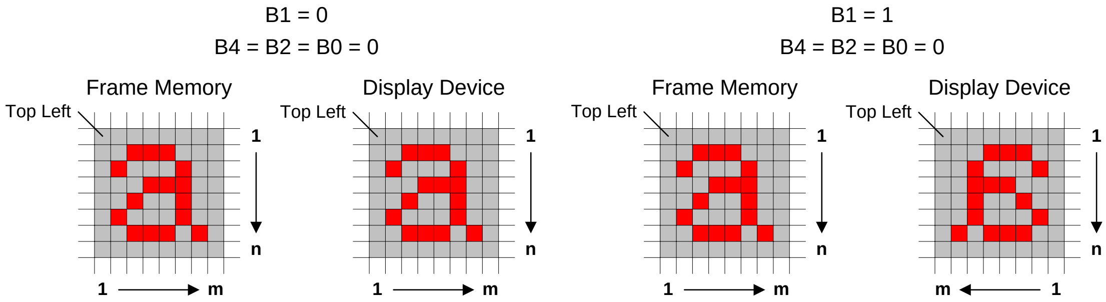


Figure 59 B1 Flip Horizontal

B0 – Flip Vertical

This bit flips the image shown on the display device top to bottom by changing the gate scanning order. Neither the frame memory contents nor the order data is read from frame memory is changed.

‘0’ = Normal

‘1’ = Flipped

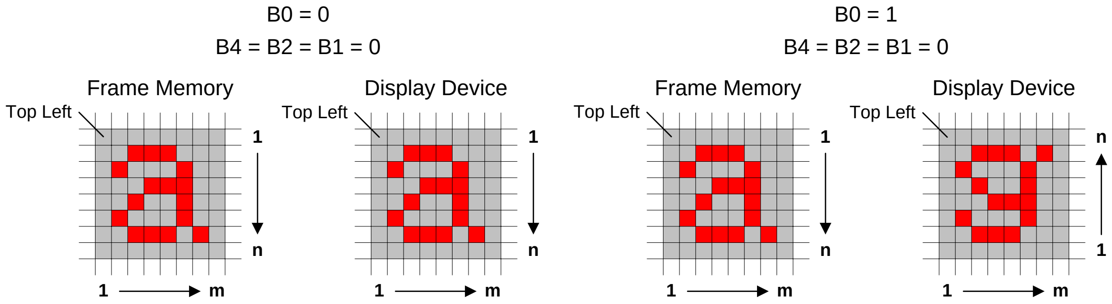


Figure 60 B0 Flip Vertical

Restrictions

None

Flow Chart

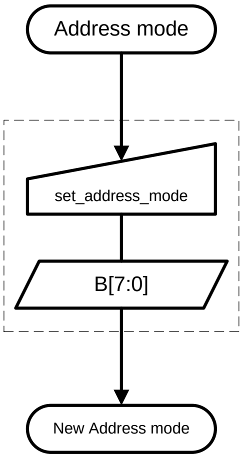


Figure 61 set_address_mode Flow Chart

6.37 set_CABC_min_brightness

Interface All

Command 5Eh

Parameters See below

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	1	1	1	0	5Eh

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	CMB15	CMB14	CMB13	CMB12	CMB11	CMB10	CMB9	CMB8	xxh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	CMB7	CMB6	CMB5	CMB4	CMB3	CMB2	CMB1	CMB0	xxh

This command sets the minimum brightness level for active CABC mode. CABC minimum brightness level step amount should follow the implementation of brightness control steps, defined in Section 6.39 (set_display_brightness).

Value 0000h means the lowest brightness level, and value FFFFh means the highest brightness level.

Note: It is up to display manufacturer to determine the implementation of this register and background logic.

Restrictions

This command is only for use with LCDs with CABC implementations. Only one parameter shall be sent for devices that support 8-bit brightness levels. Two parameters shall be sent for devices that support between 9-bit and 16-bit brightness levels.

Flow Chart

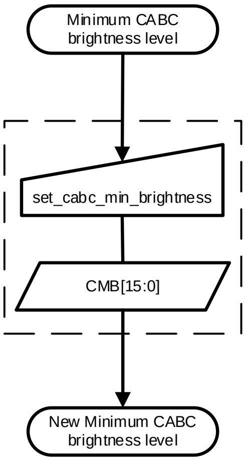

Minimum CABC brightness level
set_cabc_min_brightness
CMB[15:0]
New Minimum CABC brightness level

Figure 62 set_CABC_min_brightness Flow Chart

6.38 set_column_address

Interface All

Command 2Ah

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	0	1	0	2Ah

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	SC15	SC14	SC13	SC12	SC11	SC10	SC9	SC8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0	XXh

Parameter 3

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8	XXh

Parameter 4

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0	XXh

Description

This command defines the column extent of the frame memory accessed by the host processor with the read_memory_continue and write_memory_continue commands. No status bits are changed. A display module should not implement set_column_address in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

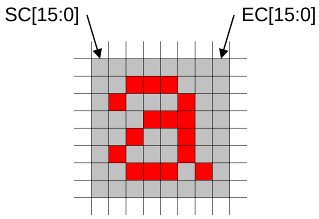


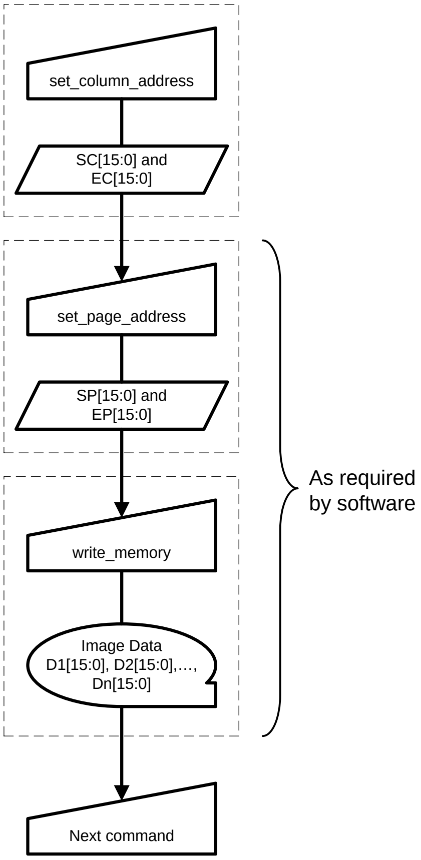
Figure 63 set_column_address Example

Restrictions

SC[15:0] must always be equal to or less than EC[15:0].

If SC[15:0] or EC[15:0] is greater than the available frame memory then the parameter is not updated.

1436 **Flow Chart**



1437

1438 **Figure 64 set_column_address Flow Chart**

6.39 set_display_brightness

Interface	All
Command	51h
Parameters	See below

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	0	0	0	1	51h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	DBV15	DBV14	DBV13	DBV12	DBV11	DBV10	DBV9	DBV8	xxh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	DBV7	DBV6	DBV5	DBV4	DBV3	DBV2	DBV1	DBV0	xxh

This command is used to adjust the brightness value of the display.

Value 0000h means the lowest brightness level, and value FFFFh means the highest brightness level.

Note: It is up to display manufacturer to determine the implementation of this register and background logic. Only one parameter shall be sent for devices that support 8-bit brightness levels. Two parameters shall be sent for devices that support between 9-bit and 16-bit brightness levels.

Restrictions

None

Flow Chart

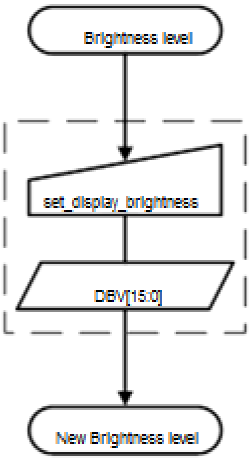


Figure 65 set_display_brightness Flow Chart

6.40 set_display_off

Interface All

Command 28h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	0	0	0	28h

Description

This command causes the display module to stop displaying the image data on the display device. The frame memory contents remain unchanged. No status bits are changed.

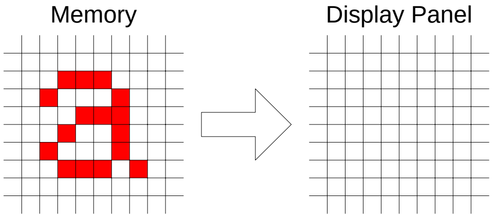


Figure 66 set_display_off Example

Restrictions

This command has no effect when the display panel is already off.

Flow Chart

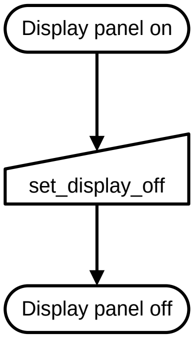


Figure 67 set_display_off Flow Chart

6.41 set_display_on

Interface All

Command 29h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	0	0	1	29h

Description

This command causes the display module to start displaying the image data on the display device. The frame memory contents remain unchanged. No status bits are changed.

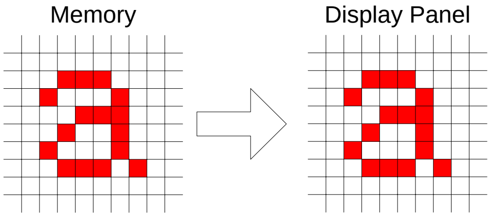


Figure 68 set_display_on Example

Restrictions

This command has no effect when the display panel is already on.

Flow Chart

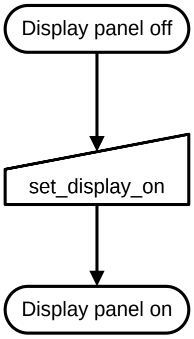


Figure 69 set_display_on Flow Chart

6.42 set_gamma_curve

Interface All

Command 26h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	0	1	1	0	26h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	GC7	GC6	GC5	GC4	GC3	GC2	GC1	GC0	XXh

Description

This command selects the desired gamma curve for the display device. Four fixed gamma curves are defined in Section 5.2. A curve is selected by setting the appropriate bit in the parameter as described in Table 10.

Table 10 Gamma Curves

GC[7:0]	Parameter	Curve Selected
00h	None	No curve selected
01h	GC0	Gamma Curve 1
02h	GC1	Gamma Curve 2
04h	GC2	Gamma Curve 3
08h	GC3	Gamma Curve 4

Note:

All other values are reserved.

Restrictions

Values of GC[7:0] not shown in Table 10 are reserved and shall not change the currently selected gamma curve.

Flow Chart

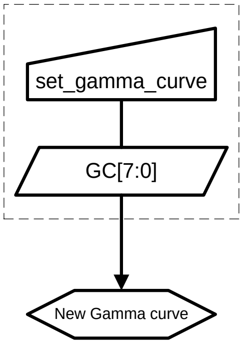


Figure 70 set_gamma_curve Flow Chart

6.43 set_page_address

Interface All

Command 2Bh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	0	1	1	2Bh

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	SP 7	SP 6	SP 5	SP 4	SP 3	SP 2	SP 1	SP 0	XXh

Parameter 3

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	XXh

Parameter 4

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	EP7	EP 6	EP 5	EP 4	EP 3	EP 2	EP 1	EP 0	XXh

Description

This command defines the page extent of the frame memory accessed by the host processor with the write_memory_continue and read_memory_continue command. No status bits are changed. A display module should not implement set_page_address in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

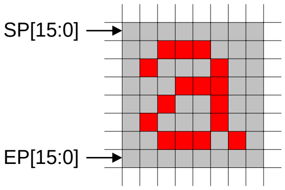


Figure 71 set_page_address Example

Restrictions

SP[15:0] must always be equal to or less than EP[15:0]

If SP[15:0] or EP[15:0] is greater than the available frame memory then the parameter is not updated.

Flow Chart

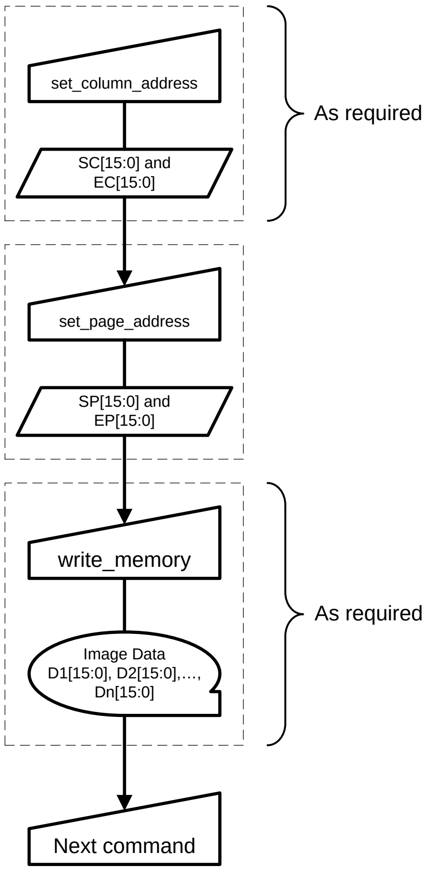


Figure 72 set_page_address Flow Chart

6.44 set_partial_columns

Interface All

Command 31h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	0	0	1	31h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	PSC15	PSC14	PSC13	PSC12	PSC11	PSC10	PSC9	PSC8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	PSC7	PSC6	PSC5	PSC4	PSC3	PSC2	PSC1	PSC0	XXh

Parameter 3

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	PEC15	PEC14	PEC13	PEC12	PEC11	PEC10	PEC9	PEC8	XXh

Parameter 4

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0	XXh

Description

This command defines the Partial Display mode’s display width. There are two parameters associated with this command, the first defines the Start Column (PSC) and the second the End Column (PEC), as illustrated in Figure 73 through Figure 76. PSC and PEC refer to the Frame Memory Column Pointer. A display module should not implement set_partial_columns in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

If End Column > Start Column

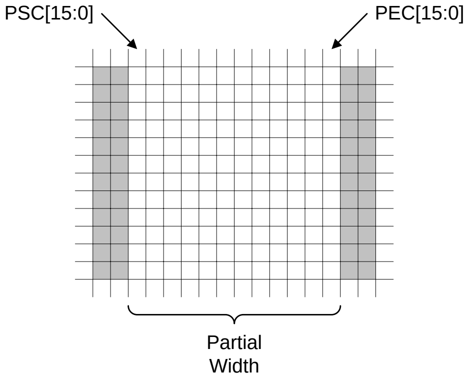


Figure 73 set_partial_columns with set_address_mode B2 = 0

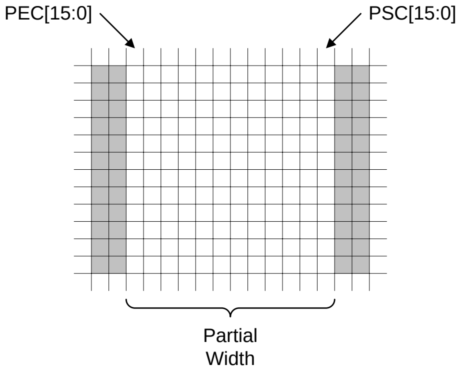


Figure 74 set_partial_columns with set_address_mode B2=1

If Start Column > End Column

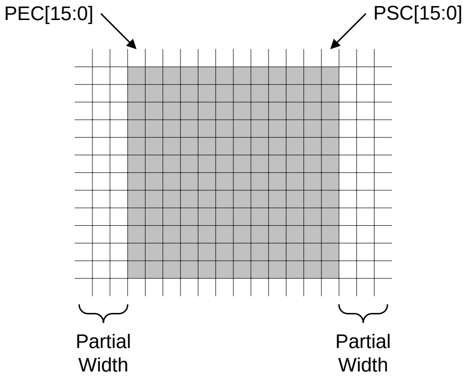


Figure 75 set_partial_columns with set_address_mode B2 = 0

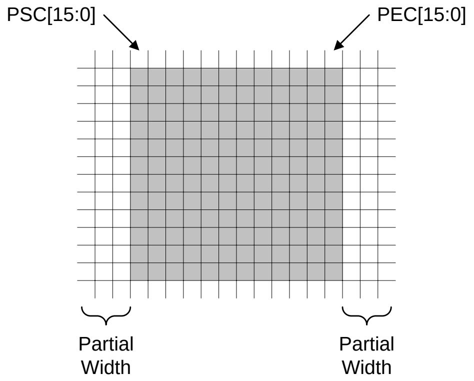


Figure 76 set_partial_columns with set_address_mode B2 = 1

Restrictions

PSC[15:0] and PEC[15:0] cannot be 0000h nor exceed the last horizontal column number.

Flow Chart

To enter Partial Display mode

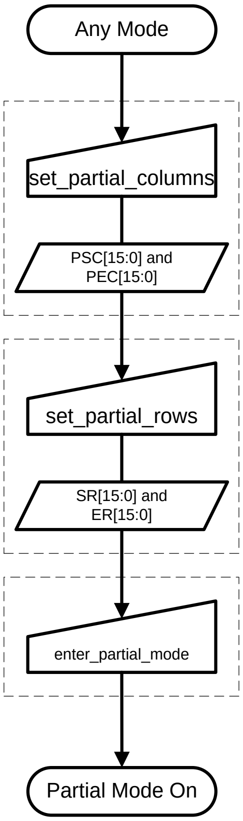


Figure 77 Entering Partial Display Mode Flow Chart

To exit Partial Display mode

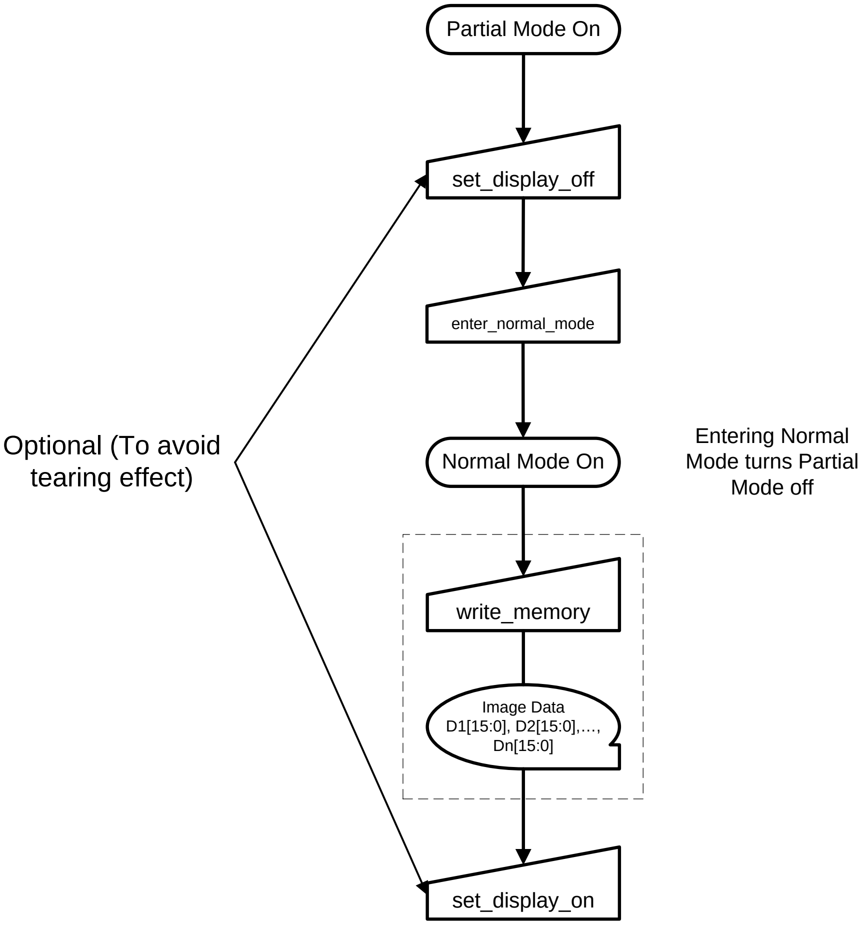

Partial Mode On
set_display_off
enter_normal_mode
Normal Mode On
Entering Normal Mode turns Partial Mode off
Optional (To avoid tearing effect)
write_memory
Image Data D1[15:0], D2[15:0],…, Dn[15:0]
set_display_on

Figure 78 Exiting Partial Display Mode Flow Chart

6.45 set_partial_rows

Interface All

Command 30h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	0	0	0	30h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0	XXh

Parameter 3

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	XXh

Parameter 4

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	XXh

Description

This command defines the Partial Display mode's display height. There are two parameters associated with this command, the first defines the Start Row (SR) and the second the End Row (ER), as illustrated in Figure 79 through Figure 82. SR and ER refer to the Frame Memory Line Pointer. A display module should not implement set_partial_rows in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

If End Row > Start Row

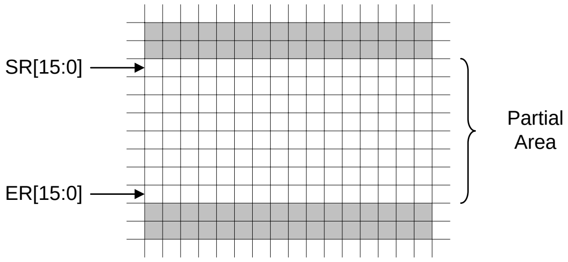


Figure 79 set_partial_rows with set_address_mode B4 = 0

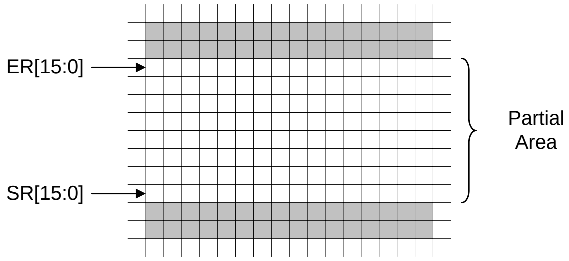


Figure 80 set_partial_rows with set_address_mode B4=1

If Start Row > End Row

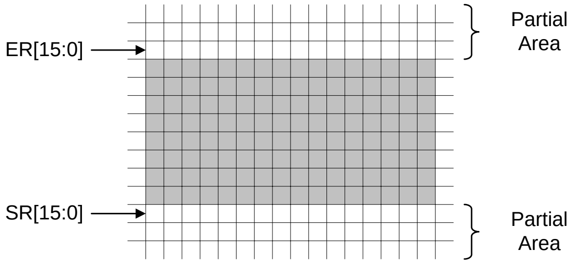


Figure 81 set_partial_rows with set_address_mode B4 = 0

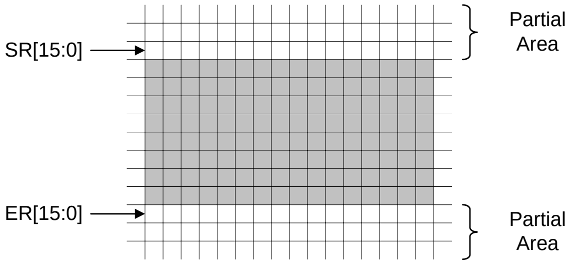


Figure 82 set_partial_rows with set_address_mode B4 = 1

Restrictions

SR[15:0] and ER[15:0] cannot be 0000h nor exceed the last vertical line number.

Flow Chart

See Section 6.44.

6.46 set_pixel_format

Interface All

Command 3Ah

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	0	1	0	3Ah

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	X	D6	D5	D4	X	D2	D1	D0	XXh

Description

This command sets the pixel format for the RGB image data used by the interface.

D[6:4] – DPI Pixel Format Definition

D[2:0] – DBI Pixel Format Definition

D7 and D3 are not used.

The pixel formats are shown in Table 7.

If a particular interface, either DBI or DPI, is not used then the corresponding bits in the parameter are ignored.

In 12, 16 and 18 bits/Pixel modes, the LUT is applied to transfer data into the frame memory.

Restrictions

There is no visible effect until the frame memory is written.

Flow Chart

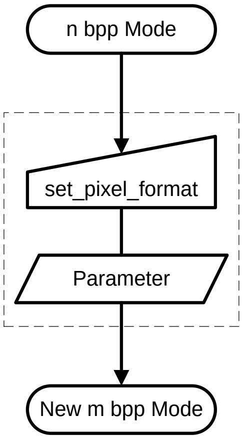


Figure 83 set_pixel_format Flow Chart

6.47 set_scroll_area

Interface All

Command 33h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	0	1	1	33h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	TFA15	TFA14	TFA13	TFA12	TFA11	TFA10	TFA9	TFA8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	TFA7	TFA6	TFA5	TFA4	TFA3	TFA2	TFA1	TFA0	XXh

Parameter 3

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	VSA15	VSA14	VSA13	VSA12	VSA11	VSA10	VSA9	VSA8	XXh

Parameter 4

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	VSA7	VSA6	VSA5	VSA4	VSA3	VSA2	VSA1	VSA0	XXh

Parameter 5

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	BFA15	BFA14	BFA13	BFA12	BFA11	BFA10	BFA9	BFA8	XXh

Parameter 6

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	BFA7	BFA6	BFA5	BFA4	BFA3	BFA2	BFA1	BFA0	XXh

Description

This command defines the display module's Vertical Scrolling Area. A display module should not implement set_scroll_area in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

If set_address_mode B4 = 0:

The 1st and 2nd parameter, TFA[15:0], describes the Top Fixed Area in number of lines from the top of the frame memory. The top of the frame memory and top of the display device are aligned.

The 3rd and 4th parameter, VSA[15:0], describes the height of the Vertical Scrolling Area in number of lines of frame memory from the Vertical Scrolling Start Address. The first line of the Vertical Scrolling Area starts immediately after the bottom most line of the Top Fixed Area. The last line of the Vertical Scrolling Area ends immediately before the top most line of the Bottom Fixed Area.

The 5th and 6th parameter, BFA[15:0], describes the Bottom Fixed Area in number of lines from the bottom of the frame memory. The bottom of the frame memory and bottom of the display device are aligned.

TFA, VSA and BFA refer to the Frame Memory Line Pointer.

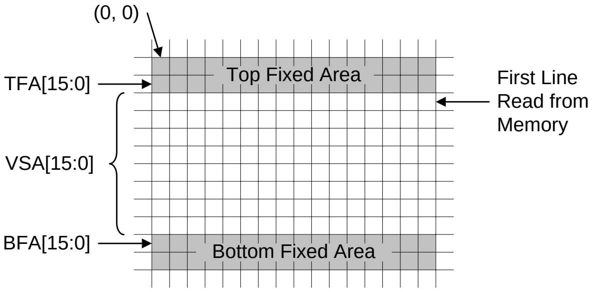


Figure 84 set_scroll_area set_address_mode B4 = 1 Example

If set_address_mode B4 = 1:

The 1st and 2nd parameter, TFA[15:0], describes the Top Fixed Area in number of lines from the bottom of the frame memory. The bottom of the frame memory and bottom of the display device are aligned.

The 3rd and 4th parameter, VSA[15:0], describes the height of the Vertical Scrolling Area in number of lines of frame memory from the Vertical Scrolling Start Address. The first line of the Vertical Scrolling Area starts immediately after the top most line of the Top Fixed Area. The last line of the Vertical Scrolling Area ends immediately before the bottom most line of the Bottom Fixed Area.

The 5th and 6th parameter, BFA[15:0], describes the Bottom Fixed Area in number of lines from the top of the frame memory. The top of the frame memory and top of the display device are aligned.

TFA, VSA and BFA refer to the Frame Memory Line Pointer.

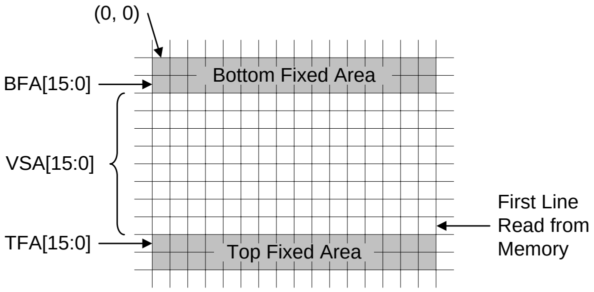


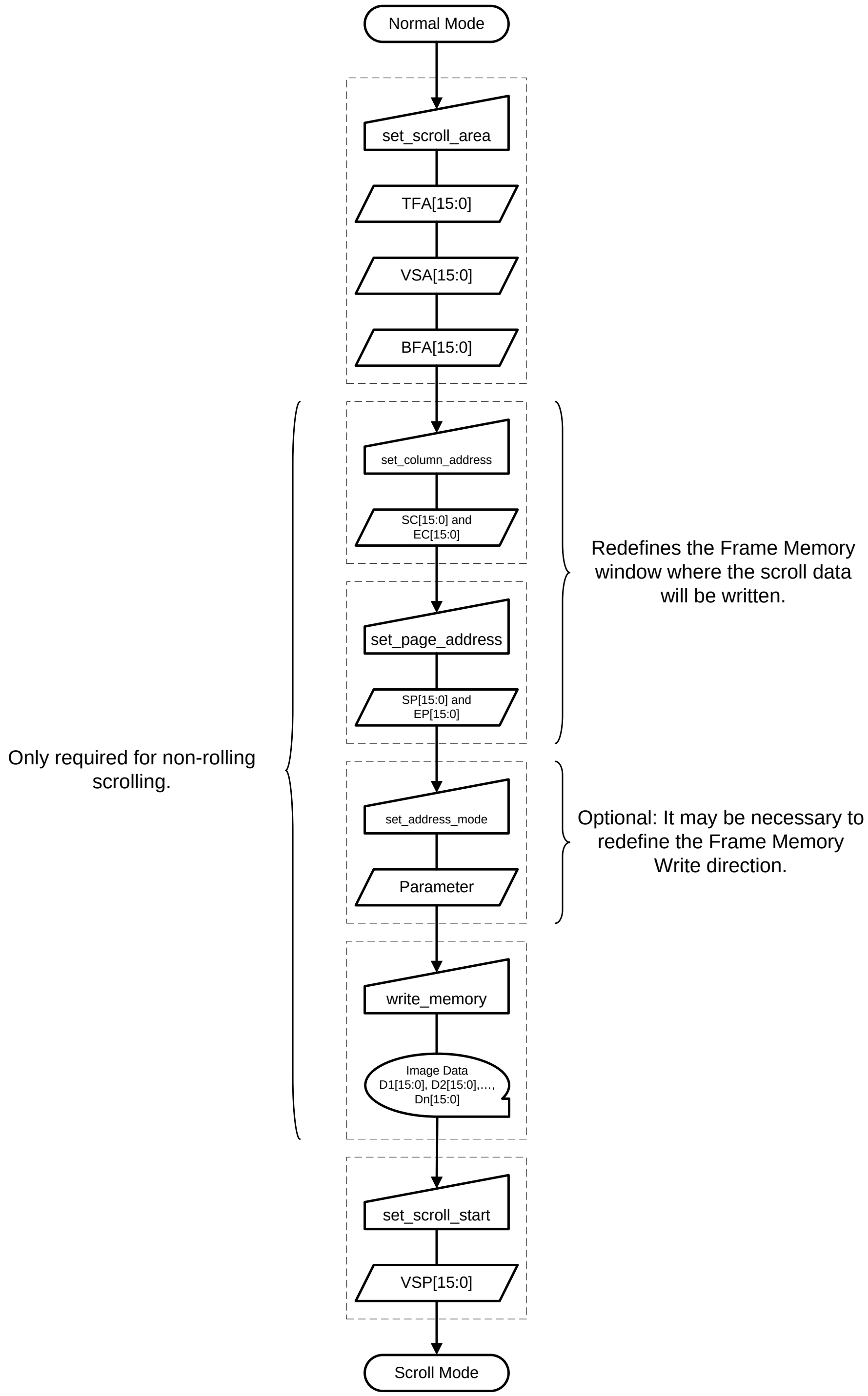
Figure 85 set_scroll_area set_address_mode B4 = 1 Example

Restrictions

The sum of TFA, VSA and BFA must equal the number of the display device’s horizontal lines (pages), otherwise Scrolling mode is undefined.

In Vertical Scroll Mode, set_address_mode B5 should be set to ‘0’ – this only affects the Frame Memory Write.

1654 **Flow Chart**


Normal Mode
set_scroll_area
TFA[15:0]
VSA[15:0]
BFA[15:0]
set_column_address
SC[15:0] and EC[15:0]
set_page_address
SP[15:0] and EP[15:0]
set_address_mode
Parameter
write_memory
Image Data D1[15:0], D2[15:0],…, Dn[15:0]
set_scroll_start
VSP[15:0]
Scroll Mode
Redefines the Frame Memory window where the scroll data will be written.
Optional: It may be necessary to redefine the Frame Memory Write direction.
Only required for non-rolling scrolling.

1655

1656 **Figure 86 set_scroll_area Flow Chart**

6.48 set_scroll_start

Interface All

Command 37h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	1	1	1	37h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	VSP15	VSP14	VSP13	VSP12	VSP11	VSP10	VSP9	VSP8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	VSP7	VSP6	VSP5	VSP4	VSP3	VSP2	VSP1	VSP0	XXh

Description

This command sets the start of the vertical scrolling area in the frame memory. The vertical scrolling area is fully defined when this command is used with the set_scroll_area command. A display module should not implement set_scroll_start in 3D Mode. If the display module implements this command in 3D Mode, the manufacturer shall specify the operation in the product datasheet.

The set_scroll_start command has one parameter, the Vertical Scroll Pointer. The VSP defines the line in the frame memory that is written to the display device as the first line of the vertical scroll area. See Section 6.47 for a description of the vertical scroll area.

The displayed image also depends on the setting of the Line Address Order bit, B4, in the set_address_mode register. See the following examples.

If set_address_mode B4 = 0:

Example:

When Top Fixed Area = Bottom Fixed Area = 0, Vertical Scrolling Area = YY and VSP = 3.

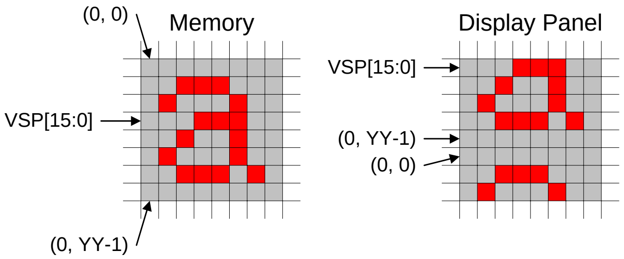


Figure 87 set_scroll_start set_address_mode B4 = 0

If set_address_mode B4 = 1:

Example:

When Top Fixed Area = Bottom Fixed Area = 0, Vertical Scrolling Area = YY and VSP = 3.

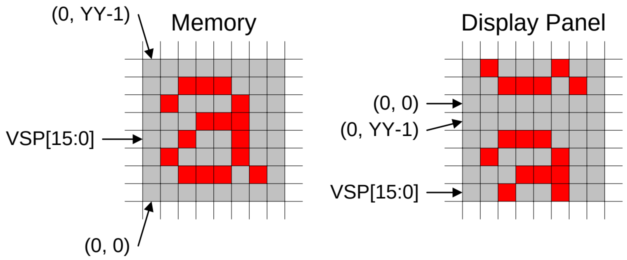


Figure 88 set_scroll_start set_address_mode B4 = 1

Restrictions

Since the value of the Vertical Scrolling Start Address is absolute with reference to the Frame Memory, it must not enter the fixed areas, see Section 6.47, otherwise an undesirable image may be shown on the Display Panel.

The following conditions shall apply:

If set_address_mode B4 = 0, TFA[15:0] - 1< VSP[15:0] < # of lines in frame memory - BFA[15:0]

If set_address_mode B4 = 1, BFA[15:0] - 1 < VSP[15:0] < # of lines in frame memory - TFA[15:0]

Flow Chart

See Section 6.47 description.

6.49 set_tear_off

Interface All

Command 34h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	1	0	0	34h

Description

This command turns off the display module’s Tearing Effect output signal on the TE signal line.

Restrictions

This command has no effect when the Tearing Effect output is already off.

Flow Chart

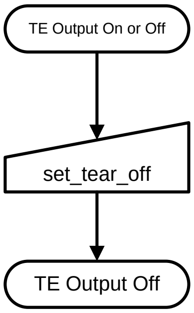


Figure 89 set_tear_off Flow Chart

6.50 set_tear_on

Interface All

Command 35h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	0	1	0	1	35h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	X	X	X	X	X	X	X	M	XXh

Description

This command turns on the display module's Tearing Effect output signal on the TE signal line. The TE signal is not affected by changing set_address_mode bit B4.

set_tear_on has one parameter that describes the Tearing Effect Output Line mode.

If M = 0 (Mode 0):

The Tearing Effect Output line consists of V-Blanking information only.

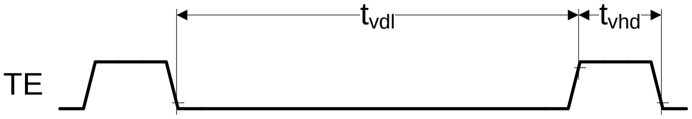


Figure 90 set_tear_on M = 0

If M = 1 (Mode 1):

The Tearing Effect Output Line consists of both V-Blanking and H-Blanking information.

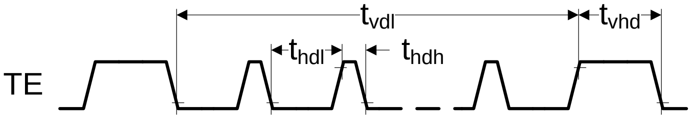


Figure 91 set_tear_on M = 1

The Tearing Effect Output line shall be active low when the display module is in Sleep mode.

See [MIPI02] for definitions of t_{vdl}, t_{vdh}, t_{hdl} and t_{hdh}.

In 3D Mode, if 3DVSYNC in set_3D_control is set to '1', a vertical sync pulse occurs between left and right images. If 3DVSYNC is set to '0', a vertical sync pulse does not occur between left and right images. 3DVSYNC shall also affect how TE pulse or TEE trigger events are issued between the left and right image data as they are scanned to the display panel.

The functionality is described by the following example:

> 3DVSYNC = '0' implies a TE sync pulse, or TEE trigger, is issued only after both left and right image data have been scanned to the display panel, regardless of the order data was sent to the display module.

3DVSYNC = ‘1’implies a TE sync pulse, or TEE trigger, is issued only after both left data scan has been finished and after right eye data has been scanned to the display panel.

See [MIPI05] for additional information.

Restrictions

This command takes affect on the frame following the current frame. Therefore, if the Tear Effect (TE) output is already ON, the TE output shall continue to operate as programmed by the previous set_tear_on, or set_tear_scanline, command until the end of the frame.

Flow Chart

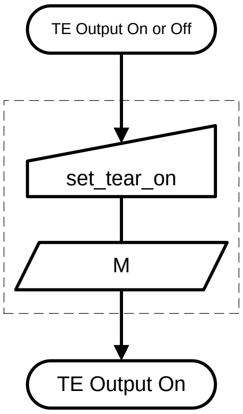


Figure 92 set_tear_on Flow Chart

6.51 set_tear_scanline

Interface All

Command 44h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	0	0	1	0	0	44h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	N15	N14	N13	N12	N11	N10	N9	N8	XXh

Parameter 2

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	N7	N6	N5	N4	N3	N2	N1	N0	XXh

Description

This command turns on the display module's Tearing Effect output signal on the TE signal line when the display module reaches line N. The TE signal is not affected by changing set_address_mode bit B4.

The Tearing Effect Line On has one parameter that describes the Tearing Effect Output Line mode.

After issuing a set_tear_scanline command to the display module, the Tearing Effect output signal, e.g. as in DBI-2 systems, shall be a delayed version of V-Blanking information as illustrated by Figure 93.

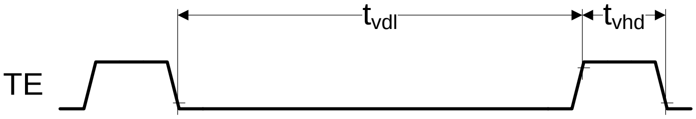


Figure 93 set_tear_scanline

Note that set_tear_scanline with N = 0 is equivalent to set_tear_on with M = 0.

The Tearing Effect Output line shall be active low when the display module is in Sleep mode.

See [MIPI02] for definitions of t_{vdl} and t_{vdh} and [MIPI03] for definition of display module line numbers.

In 2D mode, the scanline value of the display memory and the display panel is the same.

In 3D Mode, the scanline value of the display memory and the display panel can be different; set_tear_scanline shall set the scanline of the display panel.

In 3D Temporal Mode, the image input format uses top to bottom ordering. The line number shall be reset upon scanning of each frame. Thus, the host only writes the actual scan line.

Restrictions

This command takes affect on the frame following the current frame. Therefore, if the Tear Effect (TE) output is already ON, the TE output shall continue to operate as programmed by the previous set_tear_on, or set_tear_scanline, command until the end of the frame.

Flow Chart

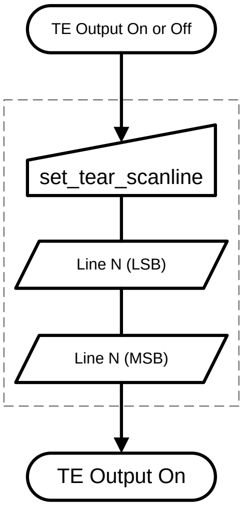


Figure 94 set_tear_scanline Flow Chart

6.52 set_vsync_timing

Interface All

Command 40h

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	0	0	0	0	0	40h

Parameter

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	RESET	DIR	LINES[4]	LINES[3]	LINES[2]	LINES[1]	LINES[0]	FRAME	XXh

Description

VSYNC is delayed or advanced by the number of scanlines in LINES, up to a maximum of thirty-two lines.

RESET – Restart display update

This bit restarts the display update. If this bit is set to ‘1’, the display module shall ignore all other bits in the parameter.

‘0’ = No operation

‘1’ = Restart display update

DIR – Line Direction

This bit determines whether VSYNC is delayed, or advanced, by the number of lines in LINES.

‘0’ = Later (Down)

‘1’ = Earlier (Up)

LINES[4:0] – Number of Lines in Adjustment

This field determines the number of lines to delay or advance VSYNC.

FRAME – Adjustment Frame

This bit determines on which frame the VSYNC adjustment is applied..

‘0’ = Next Frame

‘1’ = Frame After Next Frame

If DIR is set to ‘1’ and LINES is less than, or equal to, the number of scanlines in the VFP, a display module shall advance the start of the VSYNC by LINES scanlines. If LINES is greater than the number of scanlines in the VFP, the display module shall advance the start of VSYNC by the number of scanlines in the VFP (effectively making the VFP = 0).

If DIR is set to ‘0’ and LINES is less than, or equal to, the number of scanlines in the VBP, a display module shall delay the start of the VSYNC by LINES scanlines. If LINES is greater than the number of scanlines in the VBP, the display module shall delay the VSYNC timing by the number of scanlines in the VBP (effectively making the VBP = 0).

If FRAME is set to ‘0’, the VSYNC adjustment shall be applied to the next VSYNC.

If FRAME is set to ‘1’, the VSYNC adjustment shall be applied to the VSYNC following the next VSYNC.

If RESET is ‘1’, a display module shall restart its display panel update from pixel 1 of line 1. The display module shall also ignore all other bits in the parameter, i.e. the display module only resets the display update, it does not apply a new VSYNC adjustment when it is reset.

Restrictions

None

Flow Chart

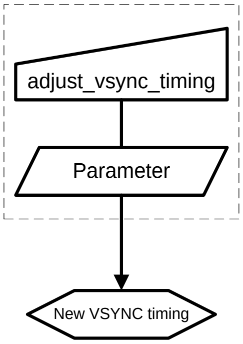


Figure 95 set_vsync_timing Flow Chart

6.53 soft_reset

Interface All

Command 01h

Parameters None

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	0	0	0	0	0	1	01h

Description

The display module performs a software reset. Registers are written with their SW Reset default values. See Section 5.7 for a list of the reset values.

Frame Memory contents are unaffected by this command.

Restrictions

The host processor must wait five milliseconds before sending any new commands to a display module following this command. The display module updates the registers during this time.

If a soft_reset is sent when the display module is in Sleep Mode, the host processor must wait 120 milliseconds before sending an exit_sleep_mode command.

soft_reset should not be sent when the display module is not in Sleep mode.

Flow Chart

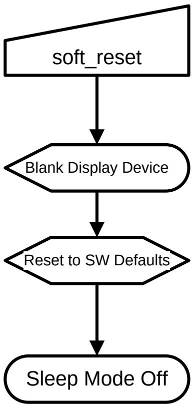


Figure 96 soft_reset Flow Chart

6.54 write_control_display

Interface All

Command 53h

Parameters See below

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	0	0	1	1	53h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	x	x	BCTRL	x	DD	BL	x	x	xxh

This command is used for brightness control mode of the display.

BCTRL – Brightness Control Block On/Off
‘0’ = Off (Brightness register is 0000h)
‘1’ = On (Brightness registers are active)

DD – Display Dimming
‘0’ = Display Dimming Off
‘1’ = Display Dimming On

BL – Backlight On/Off (For LCD)
‘0’ = Off (completely turn off backlight circuit)
‘1’ = On

Bits marked as ‘x’ are reserved for display manufacturer’s own usage.

Note: It is up to display manufacturer to determine the implementation of this register and background logic.

Restrictions

None

Flow Chart

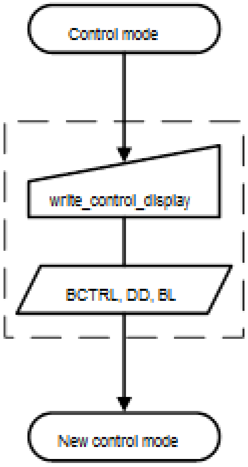


Figure 97 write_control_display Flow Chart

6.55 write_LUT

Interface All

Command 2Dh

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	1	0	1	2Dh

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	R7	R6	R5	R4	R3	R2	R1	R0	XXh

.

.

.

Parameter N

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	R7	R6	R5	R4	R3	R2	R1	R0	XXh

Parameter N + 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	G7	G6	G5	G4	G3	G2	G1	G0	XXh

.

.

.

Parameter N + M

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	G7	G6	G5	G4	G3	G2	G1	G0	XXh

Parameter N + M + 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	B7	B6	B5	B4	B3	B2	B1	B0	XXh

.

.

.

Parameter 2*N + M

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	B7	B6	B5	B4	B3	B2	B1	B0	XXh

Description

This command sets the LUT for pixel color depth conversions. Six conversions are supported as indicated in Table 11.

Table 11 LUT Color Depth Conversions

Convert from Color Depth	Convert to Color Depth		
	24	18	16
18	Yes	N/A	N/A
16	Yes	Yes	N/A
12	Yes	Yes	Yes

The LUT size depends on the pixel format of the display module. In the following list, N is the number of red or blue components and M is the number of green components in the LUT.

16-bit color display modules: N = M = 16; Total LUT Size = 2*N + M = 48 bytes.

18-bit color display modules: N = 32, M = 64; Total LUT Size = 2*N + M = 128 bytes.

24-bit color display modules: N = M = 64; Total LUT Size = 2*N + M = 192 bytes.

Regardless of host processor color depth, the defined size of the LUT shall be written according to the number of colors supported by the display module. See Annex A.

This command has no effect on other commands or the contents of frame memory. Visible changes take effect the next time the frame memory is written.

Restrictions

None

Flow Chart

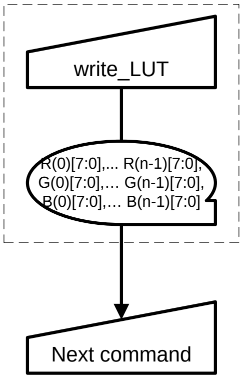


Figure 98 write_LUT Flow Chart

6.56 write_power_save

Interface All

Command 55h

Parameters See below

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	1	0	1	0	1	0	1	55h

Parameter 1

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	x	x	x	x	x	PS2	PS1	PS0	xxh

This command is used for power saving control functions of the display. These power modes could be related to CABC implementation in a LCD display.

Power saving modes are described in Table 12.

Table 12 Power Saving Modes

PS[2]	PS[1]	PS[0]	Function
0	0	0	Power Save Off
0	0	1	Power Save level: Low
0	1	0	Power Save level: Medium
0	1	1	Power Save: High
1	0	0	Outdoor mode

Bits marked as ‘x’ are reserved for display manufacturer’s own usage.

Note: It is up to display manufacturer to determine the implementation of this register and background logic.

Restrictions

None

Flow Chart

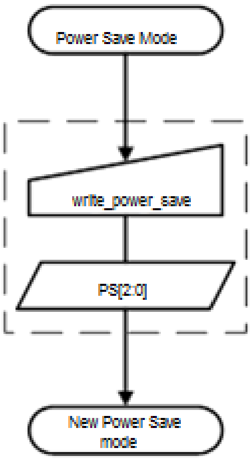


Figure 99 write_power_save Flow Chart

6.57 write_memory_continue

Interface All

Command 3Ch

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	1	1	1	0	0	3Ch

Pixel Data 1

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
H→D	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	P7	P6	P5	P4	P3	P2	P1	P0	XXh

.

.

.

Pixel Data N

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
H→D	P7	P6	P5	P4	P3	P2	P1	P0	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	P7	P6	P5	P4	P3	P2	P1	P0	XXh

Description

This command transfers image data from the host processor to the display module’s frame memory continuing from the pixel location following the previous write_memory_continue or write_memory_start command.

If set_address_mode B5 = 0:

Data is written continuing from the pixel location after the write range of the previous write_memory_start or write_memory_continue. The column register is then incremented and pixels are written to the frame memory until the column register equals the End Column (EC) value. The column register is then reset to SC and the page register is incremented. Pixels are written to the frame memory until the page register equals the End Page (EP) value and the column register equals the EC value, or the host processor sends another command. If the number of pixels exceeds (EC – SC + 1) * (EP – SP + 1) the extra pixels are ignored.

If set_address_mode B5 = 1:

Data is written continuing from the pixel location after the write range of the previous write_memory_start or write_memory_continue. The page register is then incremented and pixels are written to the frame memory until the page register equals the End Page (EP) value. The page register is then reset to SP and the column register is incremented. Pixels are written to the frame memory until the column register equals the End column (EC) value and the page register equals the EP value, or the host processor sends another command. If the number of pixels exceeds (EC – SC + 1) * (EP – SP + 1) the extra pixels are ignored.

In a DSI system, if Compression Mode bit CMODE = 1 [MIPI03] (See section 6.13):

> The Display shall treat all received pixel data as compressed image data. (See section 5.8)

See Section 6.38 for descriptions of the Start Column and End Column values.

See Section 6.43 for descriptions of the Start Page and End Page values.

See [MIPI01] and [MIPI02] for color encoding for 8 or 9 data bit image data.

Note:

> *The command description shows 16-bit pixel data transferred over a 16-bit bus. Other possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.*

The relationship between some common colors and the corresponding image data are shown in Table 13.

Table 13 Common Color Encoding

Color	Red Component	Green Component	Blue Component
Black	All bits = 0	All bits = 0	All bits = 0
Red	All bits = 1	All bits = 0	All bits = 0
Green	All bits = 0	All bits = 1	All bits = 0
Blue	All bits = 0	All bits = 0	All bits = 1
Cyan	All bits = 0	All bits = 1	All bits = 1
Yellow	All bits = 1	All bits = 1	All bits = 0
Magenta	All bits = 1	All bits = 0	All bits = 1
White	All bits = 1	All bits = 1	All bits = 1

In 3D Mode, the transmission format is defined by the set_3D_control command. The data is written into memory in the order it is received.

Restrictions

A write_memory_start should follow a set_column_address, set_page_address or set_address_mode to define the write address. Otherwise, data written with write_memory_continue is written to undefined addresses.

Flow Chart

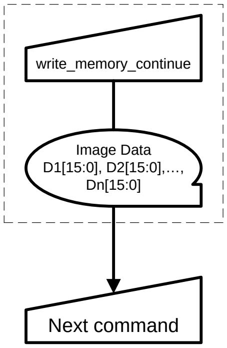


Figure 100 write_memory_continue Flow Chart

6.58 write_memory_start

Interface All

Command 2Ch

Parameters See the following description.

Command

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	0	0	1	0	1	1	0	0	2Ch

Pixel Data 1

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
H→D	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	P7	P6	P5	P4	P3	P2	P1	P0	XXh

.

.

.

Pixel Data N

Direction	D15	D14	D13	D12	D11	D10	D9	D8	Hex Code
H→D	P15	P14	P13	P12	P11	P10	P9	P8	XXh

Direction	D7	D6	D5	D4	D3	D2	D1	D0	Hex Code
H→D	P7	P6	P5	P4	P3	P2	P1	P0	XXh

Description

This command transfers image data from the host processor to the display module’s frame memory starting at the pixel location specified by preceding set_column_address and set_page_address commands (see Section 6.38 and Section 6.43).

If set_address_mode B5 = 0:

The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixel Data 1 is stored in frame memory at (SC, SP). The column register is then incremented and pixels are written to the frame memory until the column register equals the End Column (EC) value. The column register is then reset to SC and the page register is incremented. Pixels are written to the frame memory until the page register equals the End Page (EP) value and the column register equals the EC value, or the host processor sends another command. If the number of pixels exceeds (EC – SC + 1) * (EP – SP + 1) the extra pixels are ignored.

If set_address_mode B5 = 1:

The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

Pixel Data 1 is stored in frame memory at (SC, SP). The page register is then incremented and pixels are written to the frame memory until the page register equals the End Page (EP) value. The page register is then reset to SP and the column register is incremented. Pixels are written to the frame memory until the column register equals the End column (EC) value and the page register equals the EP value, or the host processor sends another command. If the number of pixels exceeds (EC – SC + 1) * (EP – SP + 1) the extra pixels are ignored.

In a DSI system, if Compression Mode bit CMODE = 1 [MIPI03] (See section 6.13):

The Display shall treat all received pixel data as compressed image data. (See section 5.8)

See Section 6.38 for descriptions of the Start Column and End Column values.

See Section 6.43 for descriptions of the Start Page and End Page values.

See [MIPI01] and [MIPI02] for color encoding for 8 or 9 data bit image data.

Note:

The command description shows 16-bit pixel data transferred over a 16-bit bus. Other possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.

The relationship between some common colors and the corresponding image data are shown in Table 14.

Table 14 Common Color Encoding

Color	Red Component	Green Component	Blue Component
Black	All bits = 0	All bits = 0	All bits = 0
Red	All bits = 1	All bits = 0	All bits = 0
Green	All bits = 0	All bits = 1	All bits = 0
Blue	All bits = 0	All bits = 0	All bits = 1
Cyan	All bits = 0	All bits = 1	All bits = 1
Yellow	All bits = 1	All bits = 1	All bits = 0
Magenta	All bits = 1	All bits = 0	All bits = 1
White	All bits = 1	All bits = 1	All bits = 1

In 3D Mode, the transmission format is defined by the set_3D_control command. The data is written into memory in the order it is received.

Restrictions

A write_memory_start should follow a set_column_address, set_page_address or set_address_mode to define the write location. Otherwise, data written with write_memory_start and any following write_memory_continue commands is written to undefined locations.

Flow Chart

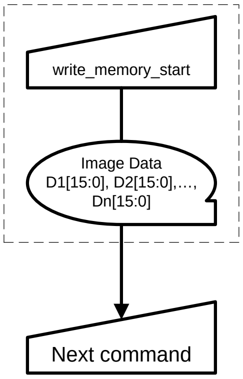


Figure 101 write_memory_start Flow Chart

Annex A Pixel-to-Byte Mapping

Many of the commands in this specification utilize display panel properties and therefore refer to pixels and scan lines. However, numerous components of a display system are inherently byte oriented. Therefore, a consistent method should be used to convert pixel formats to bytes to ensure interoperability among all components. This Section defines the pixel-to-byte mapping used by this specification.

Note:

> *The set_address_mode command (Section 6.35) affects the bit ordering within a pixel, red and blue components may be swapped, and the order pixels are transferred across the interface. The figures in this section are shown with set_address_mode B4=B5=B6=B7=0.*

A.1 Three Bits per Pixel Format

Three bits per pixel formats do not map directly to byte boundaries and therefore require special handling. In this pixel format, each byte holds two pixels. Two bits in each byte convey no color information. The organization of bits is shown in Figure 102.

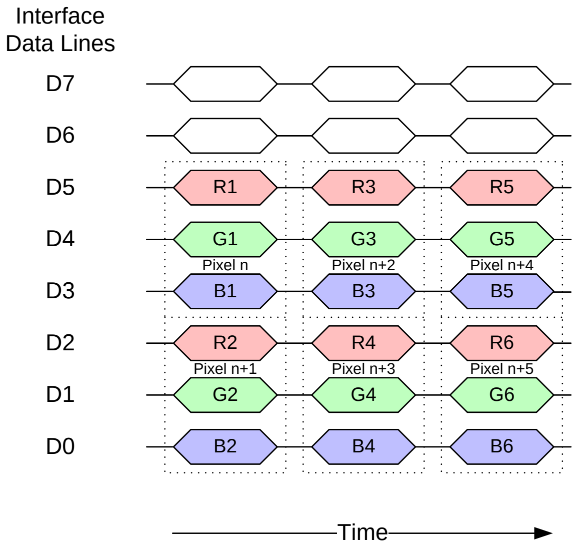


Figure 102 Three Bits per Pixel Format to Byte Mapping

A.2 Eight Bits per Pixel Format

Eight bits per pixel formats map directly to byte boundaries and therefore require no special handling. Figure 103 shows the mapping of pixels to bytes.

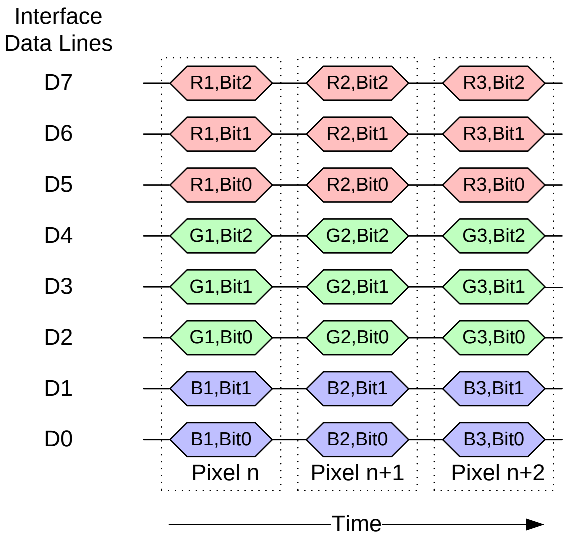


Figure 103 Eight Bits per Pixel Format to Byte Mapping

A.3 Twelve Bits per Pixel Format

Twelve bits per pixel formats do not map directly to byte boundaries and therefore require special handling. In this pixel format, three bytes hold two pixels. Figure 104 shows the mapping of pixels to bytes.

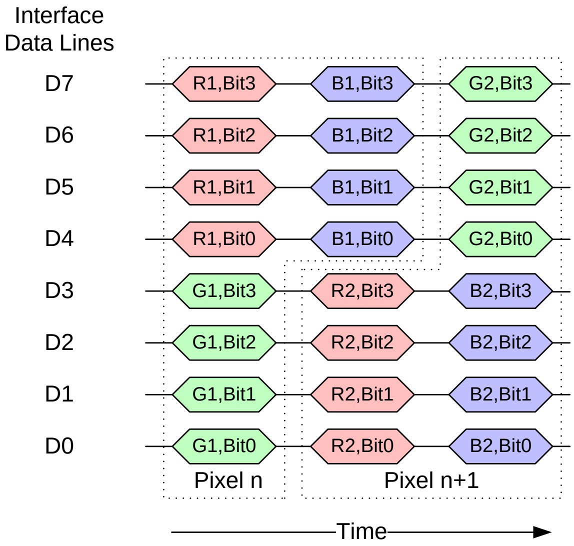


Figure 104 Twelve Bits per Pixel Format to Byte Mapping

With this format, pixel boundaries align with byte boundaries every two pixels (three bytes). For pixel data, the total line width should be a multiple of three bytes. However, the value in WC (size of payload in bytes) shall not be restricted to non-zero values divisible by three.

A.4 Sixteen Bits per Pixel Format

Sixteen bits per pixel formats do not map directly to byte boundaries and therefore require special handling. However, this format is simpler than twelve bit formats since one pixel occupies two bytes. Figure 105 shows the mapping of pixels to bytes.

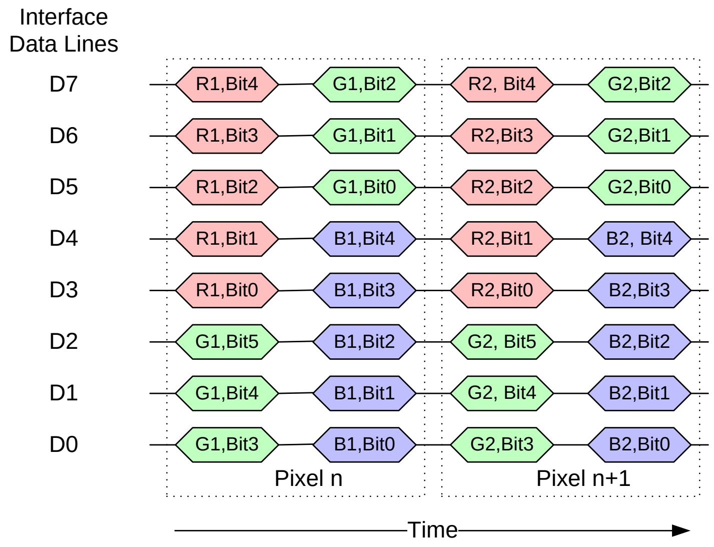


Figure 105 Sixteen Bits per Pixel Format to Byte Mapping

A.5 Eighteen Bits per Pixel Format

Eighteen bits per pixel formats do not map directly to byte boundaries and therefore require special handling. In this pixel format, each pixel occupies three bytes (24-bits), one for each color component. Two bits in each byte convey no color information. Figure 106 shows the mapping of pixels to bytes.

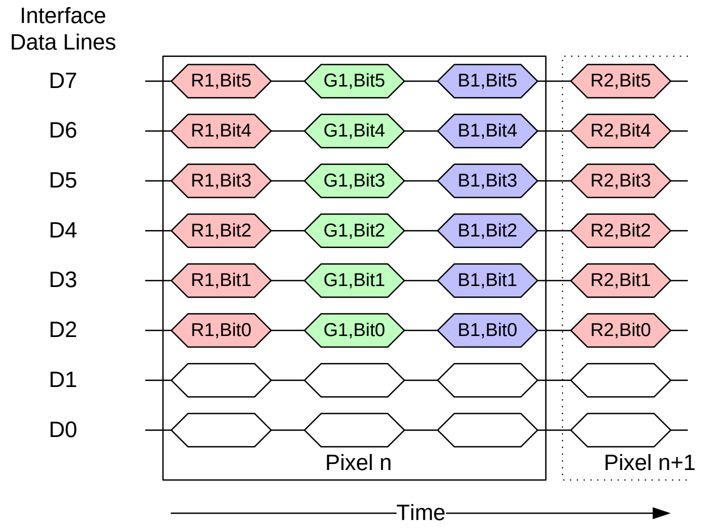


Figure 106 Eighteen Bits per Pixel Format to Byte Mapping

A.6 Twenty-Four Bits per Pixel Format

Twenty-four bits per pixel formats do not map directly to byte boundaries and therefore require special handling. This format is similar to the eighteen bits per pixel format since one pixel occupies three bytes. However, all bits in this format convey color information. Figure 107 shows the mapping of pixels to bytes.

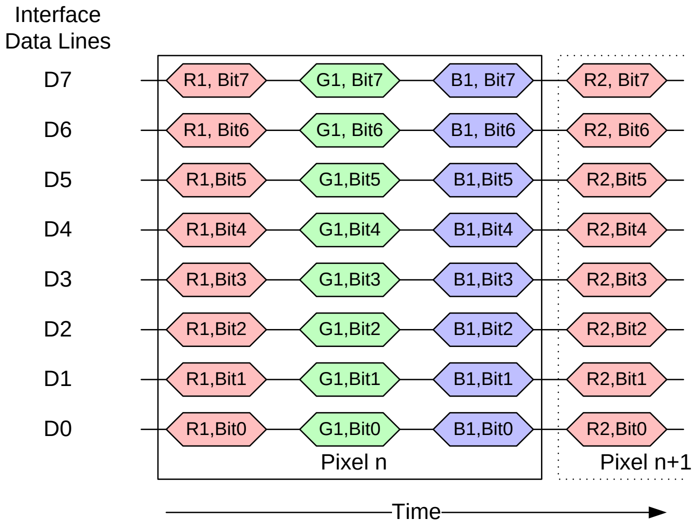
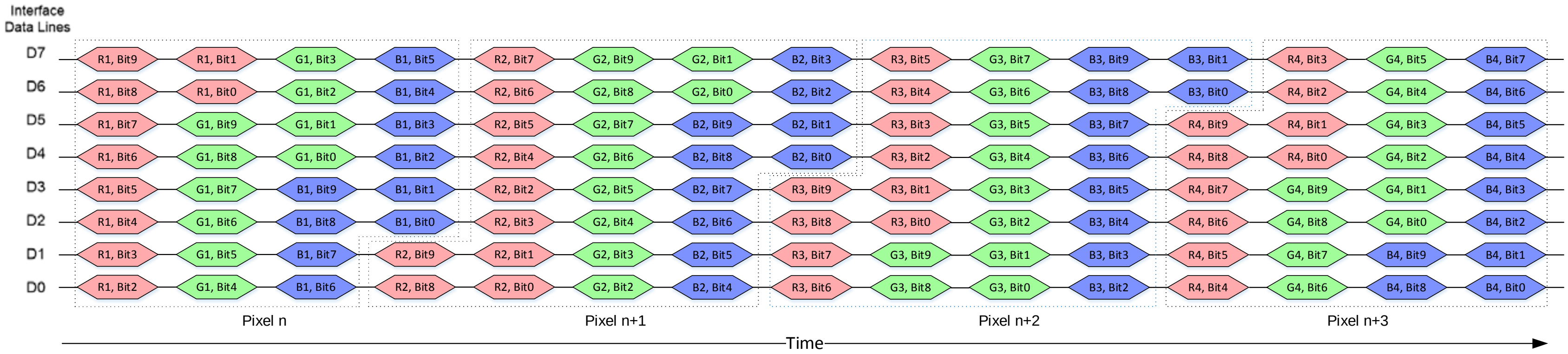


Figure 107 Twenty-Four Bits per Pixel Format to Byte Mapping

2062
A.7 Thirty Bits per Pixel Format

2063 Thirty bits per pixel formats do not map directly to byte boundaries and therefore require special handling. In this pixel format, fifteen bytes hold four pixels
2064 tightly packed. Figure 108 shows the mapping of pixels to bytes.



D7	R1, Bit9	R1, Bit1	G1, Bit3	B1, Bit5	R2, Bit7	G2, Bit9	G2, Bit1	B2, Bit3	R3, Bit5	G3, Bit7	B3, Bit9	B3, Bit1	R4, Bit3	G4, Bit5	B4, Bit7
D6	R1, Bit8	R1, Bit0	G1, Bit2	B1, Bit4	R2, Bit6	G2, Bit8	G2, Bit0	B2, Bit2	R3, Bit4	G3, Bit6	B3, Bit8	B3, Bit0	R4, Bit2	G4, Bit4	B4, Bit6
D5	R1, Bit7	G1, Bit9	G1, Bit1	B1, Bit3	R2, Bit5	G2, Bit7	B2, Bit9	B2, Bit1	R3, Bit3	G3, Bit5	B3, Bit7	R4, Bit9	R4, Bit1	G4, Bit3	B4, Bit5
D4	R1, Bit6	G1, Bit8	G1, Bit0	B1, Bit2	R2, Bit4	G2, Bit6	B2, Bit8	B2, Bit0	R3, Bit2	G3, Bit4	B3, Bit6	R4, Bit8	R4, Bit0	G4, Bit2	B4, Bit4
D3	R1, Bit5	G1, Bit7	B1, Bit9	B1, Bit1	R2, Bit2	G2, Bit5	B2, Bit7	R3, Bit9	R3, Bit1	G3, Bit3	B3, Bit5	R4, Bit7	G4, Bit9	G4, Bit1	B4, Bit3
D2	R1, Bit4	G1, Bit6	B1, Bit8	B1, Bit0	R2, Bit3	G2, Bit4	B2, Bit6	R3, Bit8	R3, Bit0	G3, Bit2	B3, Bit4	R4, Bit6	G4, Bit8	G4, Bit0	B4, Bit2
D1	R1, Bit3	G1, Bit5	B1, Bit7	R2, Bit9	R2, Bit1	G2, Bit3	B2, Bit5	R3, Bit7	G3, Bit9	G3, Bit1	B3, Bit3	R4, Bit5	G4, Bit7	B4, Bit9	B4, Bit1
D0	R1, Bit2	G1, Bit4	B1, Bit6	R2, Bit8	R2, Bit0	G2, Bit2	B2, Bit4	R3, Bit6	G3, Bit8	G3, Bit0	B3, Bit2	R4, Bit4	G4, Bit6	B4, Bit8	B4, Bit0

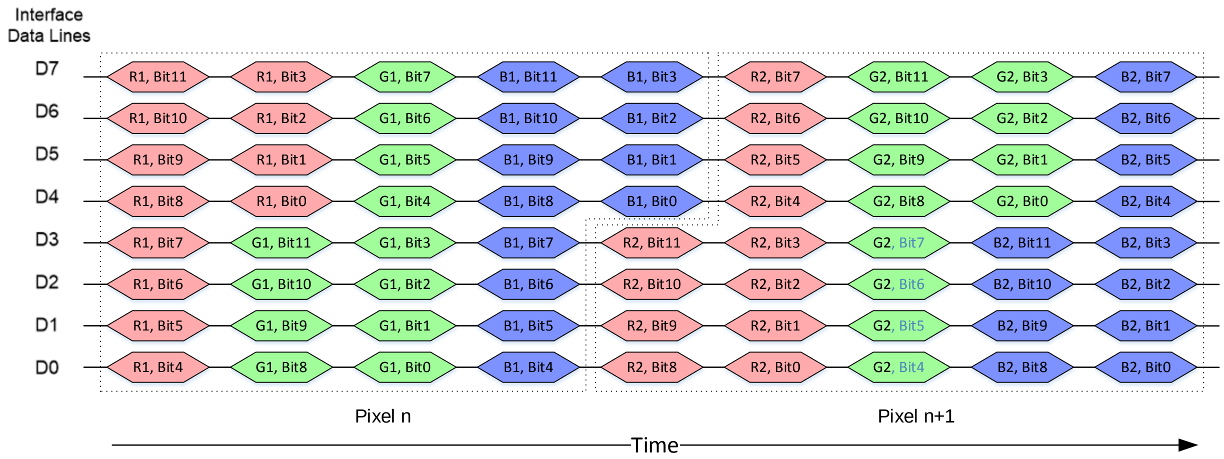
2065

2066 **Figure 108 Thirty Bits per Pixel Format to Byte Mapping**

2067 With this format, pixel boundaries align with byte boundaries every four pixels (fifteen bytes). For pixel data, the total line width should be a multiple of fifteen
2068 bytes. However, the value in WC (size of payload in bytes) shall not be restricted to non-zero values divisible by fifteen.

A.8 Thirty-Six Bits per Pixel Format

Thirty-six bits per pixel formats do not map directly to byte boundaries and therefore require special handling. In this pixel format, nine bytes hold two pixels tightly packed. Figure 109 shows the mapping of pixels to bytes.



D7	R1, Bit11	R1, Bit3	G1, Bit7	B1, Bit11	B1, Bit3	R2, Bit7	G2, Bit11	G2, Bit3	B2, Bit7
D6	R1, Bit10	R1, Bit2	G1, Bit6	B1, Bit10	B1, Bit2	R2, Bit6	G2, Bit10	G2, Bit2	B2, Bit6
D5	R1, Bit9	R1, Bit1	G1, Bit5	B1, Bit9	B1, Bit1	R2, Bit5	G2, Bit9	G2, Bit1	B2, Bit5
D4	R1, Bit8	R1, Bit0	G1, Bit4	B1, Bit8	B1, Bit0	R2, Bit4	G2, Bit8	G2, Bit0	B2, Bit4
D3	R1, Bit7	G1, Bit11	G1, Bit3	B1, Bit7	R2, Bit11	R2, Bit3	G2, Bit7	B2, Bit11	B2, Bit3
D2	R1, Bit6	G1, Bit10	G1, Bit2	B1, Bit6	R2, Bit10	R2, Bit2	G2, Bit6	B2, Bit10	B2, Bit2
D1	R1, Bit5	G1, Bit9	G1, Bit1	B1, Bit5	R2, Bit9	R2, Bit1	G2, Bit5	B2, Bit9	B2, Bit1
D0	R1, Bit4	G1, Bit8	G1, Bit0	B1, Bit4	R2, Bit8	R2, Bit0	G2, Bit4	B2, Bit8	B2, Bit0

Figure 109 Thirty-Six Bits per Pixel Format to Byte Mapping

With this format, pixel boundaries align with byte boundaries every two pixels (nine bytes). For pixel data, the total line width should be a multiple of nine bytes. However, the value in WC (size of payload in bytes) shall not be restricted to non-zero values divisible by nine.

Annex B Color Depth Conversion Look-up Tables (informative)

B.1 Color Depth Conversion LUT – 12-bit Color to 16-bit Color

Table 15 12-bit to 16-bit LUT Red Component Values

R input (4-bit)	R output (5-bit)	
12-bits/pixel	16-bits/pixel	
4,096 colors	65,536 colors	write_LUT Parameter
0000	R_{004} R_{003} R_{002} R_{001} R_{000}	1
0001	R_{014} R_{013} R_{012} R_{011} R_{010}	2
0010	R_{024} R_{023} R_{022} R_{021} R_{020}	3
0011	R_{034} R_{033} R_{032} R_{031} R_{030}	4
0100	R_{044} R_{043} R_{042} R_{041} R_{040}	5
0101	R_{054} R_{053} R_{052} R_{051} R_{050}	6
0110	R_{064} R_{063} R_{062} R_{061} R_{060}	7
0111	R_{074} R_{073} R_{072} R_{071} R_{070}	8
1000	R_{084} R_{083} R_{082} R_{081} R_{080}	9
1001	R_{094} R_{093} R_{092} R_{091} R_{090}	10
1010	R_{104} R_{103} R_{102} R_{101} R_{100}	11
1011	R_{114} R_{113} R_{112} R_{111} R_{110}	12
1100	R_{124} R_{123} R_{122} R_{121} R_{120}	13
1101	R_{134} R_{133} R_{132} R_{131} R_{130}	14
1110	R_{144} R_{143} R_{142} R_{141} R_{140}	15
1111	R_{154} R_{153} R_{152} R_{151} R_{150}	16

Table 16 12-bit to 16-bit LUT Green Component Values

G input (4bit) 12 bit/pixel -mode 4,096 colors	G output (6bit) 16 bit/pixel -mode 65,536 colors	write_LUT Parameter
0000	G_{005} G_{004} G_{003} G_{002} G_{001} G_{000}	17
0001	G_{015} G_{014} G_{013} G_{012} G_{011} G_{010}	18
0010	G_{025} G_{024} G_{023} G_{022} G_{021} G_{020}	19
0011	G_{035} G_{034} G_{033} G_{032} G_{031} G_{030}	20
0100	G_{045} G_{044} G_{043} G_{042} G_{041} G_{040}	21
0101	G_{055} G_{054} G_{053} G_{052} G_{051} G_{050}	22
0110	G_{065} G_{064} G_{063} G_{062} G_{061} G_{060}	23
0111	G_{075} G_{074} G_{073} G_{072} G_{071} G_{070}	24
1000	G_{085} G_{084} G_{083} G_{082} G_{081} G_{080}	25
1001	G_{095} G_{094} G_{093} G_{092} G_{091} G_{090}	26
1010	G_{105} G_{104} G_{103} G_{102} G_{101} G_{100}	27
1011	G_{115} G_{114} G_{113} G_{112} G_{111} G_{110}	28
1100	G_{125} G_{124} G_{123} G_{122} G_{121} G_{120}	29
1101	G_{135} G_{134} G_{133} G_{132} G_{131} G_{130}	30
1110	G_{145} G_{144} G_{143} G_{142} G_{141} G_{140}	31
1111	G_{155} G_{154} G_{153} G_{152} G_{151} G_{150}	32

Table 17 12-bit to 16-bit LUT Blue Component Values

B input (4bit) 12 bit/pixel -mode 4,096 colors	B output (5bit) 16 bit/pixel -mode 65,536 colors	write_LUT Parameter
0000	B_{004} B_{003} B_{002} B_{001} B_{000}	33
0001	B_{014} B_{013} B_{012} B_{011} B_{010}	34
0010	B_{024} B_{023} B_{022} B_{021} B_{020}	35
0011	B_{034} B_{033} B_{032} B_{031} B_{030}	36
0100	B_{044} B_{043} B_{042} B_{041} B_{040}	37
0101	B_{054} B_{053} B_{052} B_{051} B_{050}	38
0110	B_{064} B_{063} B_{062} B_{061} B_{060}	39
0111	B_{074} B_{073} B_{072} B_{071} B_{070}	40
1000	B_{084} B_{083} B_{082} B_{081} B_{080}	41
1001	B_{094} B_{093} B_{092} B_{091} B_{090}	42
1010	B_{104} B_{103} B_{102} B_{101} B_{100}	43
1011	B_{114} B_{113} B_{112} B_{111} B_{110}	44
1100	B_{124} B_{123} B_{122} B_{121} B_{120}	45
1101	B_{134} B_{133} B_{132} B_{131} B_{130}	46
1110	B_{144} B_{143} B_{142} B_{141} B_{140}	47
1111	B_{154} B_{153} B_{152} B_{151} B_{150}	48

B.2 Color Depth Conversion LUT – 12-bit and 16-bit Colors to 18-bit Color

Table 18 12-bit, 16-bit to 18-bit LUT Red Component Values

R input (4bit) 12 bit/pixel -mode 4,096 colors	R input (5 bit) 16 bit/pixel -mode 65,536 colors	R output (6bit) 18 bit/pixel -mode 262,144 colors	write_LUT Parameter
0000	00000	R_{005} R_{004} R_{003} R_{002} R_{001} R_{000}	1
0001	00001	R_{015} R_{014} R_{013} R_{012} R_{011} R_{010}	2
0010	00010	R_{025} R_{024} R_{023} R_{022} R_{021} R_{020}	3
0011	00011	R_{035} R_{034} R_{033} R_{032} R_{031} R_{030}	4
0100	00100	R_{045} R_{044} R_{043} R_{042} R_{041} R_{040}	5
0101	00101	R_{055} R_{054} R_{053} R_{052} R_{051} R_{050}	6
0110	00110	R_{065} R_{064} R_{063} R_{062} R_{061} R_{060}	7
0111	00111	R_{075} R_{074} R_{073} R_{072} R_{071} R_{070}	8
1000	01000	R_{085} R_{084} R_{083} R_{082} R_{081} R_{080}	9
1001	01001	R_{095} R_{094} R_{093} R_{092} R_{091} R_{090}	10
1010	01010	R_{105} R_{104} R_{103} R_{102} R_{101} R_{100}	11
1011	01011	R_{115} R_{114} R_{113} R_{112} R_{111} R_{110}	12
1100	01100	R_{125} R_{124} R_{123} R_{122} R_{121} R_{120}	13
1101	01101	R_{135} R_{134} R_{133} R_{132} R_{131} R_{130}	14
1110	01110	R_{145} R_{144} R_{143} R_{142} R_{141} R_{140}	15
1111	01111	R_{155} R_{154} R_{153} R_{152} R_{151} R_{150}	16
No Input	10000	R_{165} R_{164} R_{163} R_{162} R_{161} R_{160}	17
No Input	10001	R_{175} R_{174} R_{173} R_{172} R_{171} R_{170}	18
No Input	10010	R_{185} R_{184} R_{183} R_{182} R_{181} R_{180}	19
No Input	10011	R_{195} R_{194} R_{193} R_{192} R_{191} R_{190}	20
No Input	10100	R_{205} R_{204} R_{203} R_{202} R_{201} R_{200}	21
No Input	10101	R_{215} R_{214} R_{213} R_{212} R_{211} R_{210}	22
No Input	10110	R_{225} R_{224} R_{223} R_{222} R_{221} R_{220}	23
No Input	10111	R_{235} R_{234} R_{233} R_{232} R_{231} R_{230}	24
No Input	11000	R_{245} R_{244} R_{243} R_{242} R_{241} R_{240}	25
No Input	11001	R_{255} R_{254} R_{253} R_{252} R_{251} R_{250}	26
No Input	11010	R_{265} R_{264} R_{263} R_{262} R_{261} R_{260}	27
No Input	11011	R_{275} R_{274} R_{273} R_{272} R_{271} R_{270}	28
No Input	11100	R_{285} R_{284} R_{283} R_{282} R_{281} R_{280}	29
No Input	11101	R_{295} R_{294} R_{293} R_{292} R_{291} R_{290}	30
No Input	11110	R_{305} R_{304} R_{303} R_{302} R_{301} R_{300}	31
No Input	11111	R_{315} R_{314} R_{313} R_{312} R_{311} R_{310}	32

Table 19 12-bit, 16-bit to 18-bit LUT Green Component Values

G input (4bit) 12 bit/pixel -mode 4,096 colors	G input (6 bit) 16 bit/pixel -mode 65,536 colors	G output (6bit) 18 bit/pixel -mode 262,144 colors	write_LUT Parameter
0000	000000	G_{005} G_{004} G_{003} G_{002} G_{001} G_{000}	33
0001	000001	G_{015} G_{014} G_{013} G_{012} G_{011} G_{010}	34
0010	000010	G_{025} G_{024} G_{023} G_{022} G_{021} G_{020}	35
0011	000011	G_{035} G_{034} G_{033} G_{032} G_{031} G_{030}	36
0100	000100	G_{045} G_{044} G_{043} G_{042} G_{041} G_{040}	37
0101	000101	G_{055} G_{054} G_{053} G_{052} G_{051} G_{050}	38
0110	000110	G_{065} G_{064} G_{063} G_{062} G_{061} G_{060}	39
0111	000111	G_{075} G_{074} G_{073} G_{072} G_{071} G_{070}	40
1000	001000	G_{085} G_{084} G_{083} G_{082} G_{081} G_{080}	41
1001	001001	G_{095} G_{094} G_{093} G_{092} G_{091} G_{090}	42
1010	001010	G_{105} G_{104} G_{103} G_{102} G_{101} G_{100}	43
1011	001011	G_{115} G_{114} G_{113} G_{112} G_{111} G_{110}	44
1100	001100	G_{125} G_{124} G_{123} G_{122} G_{121} G_{120}	45
1101	001101	G_{135} G_{134} G_{133} G_{132} G_{131} G_{130}	46
1110	001110	G_{145} G_{144} G_{143} G_{142} G_{141} G_{140}	47
1111	001111	G_{155} G_{154} G_{153} G_{152} G_{151} G_{150}	48
No Input	010000	G_{165} G_{164} G_{163} G_{162} G_{161} G_{160}	49
No Input	010001	G_{175} G_{174} G_{173} G_{172} G_{171} G_{170}	50
No Input	010010	G_{185} G_{184} G_{183} G_{182} G_{181} G_{180}	51
No Input	010011	G_{195} G_{194} G_{193} G_{192} G_{191} G_{190}	52
No Input	010100	G_{205} G_{204} G_{203} G_{202} G_{201} G_{200}	53
No Input	010101	G_{215} G_{214} G_{213} G_{212} G_{211} G_{210}	54
No Input	010110	G_{225} G_{224} G_{223} G_{222} G_{221} G_{220}	55
No Input	010111	G_{235} G_{234} G_{233} G_{232} G_{231} G_{230}	56
No Input	011000	G_{245} G_{244} G_{243} G_{242} G_{241} G_{240}	57
No Input	011001	G_{255} G_{254} G_{253} G_{252} G_{251} G_{250}	58
No Input	011010	G_{265} G_{264} G_{263} G_{262} G_{261} G_{260}	59
No Input	011011	G_{275} G_{274} G_{273} G_{272} G_{271} G_{270}	60
No Input	011100	G_{285} G_{284} G_{283} G_{282} G_{281} G_{280}	61
No Input	011101	G_{295} G_{294} G_{293} G_{292} G_{291} G_{290}	62
No Input	011110	G_{305} G_{304} G_{303} G_{302} G_{301} G_{300}	63
No Input	011111	G_{315} G_{314} G_{313} G_{312} G_{311} G_{310}	64
No Input	100000	G_{325} G_{324} G_{323} G_{322} G_{321} G_{320}	65
No Input	100001	G_{335} G_{334} G_{333} G_{332} G_{331} G_{330}	66

G input (4bit) **12 bit/pixel -mode** **4,096 colors**	**G input (6 bit)** **16 bit/pixel -mode** **65,536 colors**	**G output (6bit)** **18 bit/pixel -mode** **262,144 colors**	**write_LUT** **Parameter**
No Input	100010	G_{345} G_{344} G_{343} G_{342} G_{341} G_{340}	67
No Input	100011	G_{355} G_{354} G_{353} G_{352} G_{351} G_{350}	68
No Input	100100	G_{365} G_{364} G_{363} G_{362} G_{361} G_{360}	69
No Input	100101	G_{375} G_{374} G_{373} G_{372} G_{371} G_{370}	70
No Input	100110	G_{385} G_{384} G_{383} G_{382} G_{381} G_{380}	71
No Input	100111	G_{395} G_{394} G_{393} G_{392} G_{391} G_{390}	72
No Input	101000	G_{405} G_{404} G_{403} G_{402} G_{401} G_{400}	73
No Input	101001	G_{415} G_{414} G_{413} G_{412} G_{411} G_{410}	74
No Input	101010	G_{425} G_{424} G_{423} G_{422} G_{421} G_{420}	75
No Input	101011	G_{435} G_{434} G_{433} G_{432} G_{431} G_{430}	76
No Input	101100	G_{445} G_{444} G_{443} G_{442} G_{441} G_{440}	77
No Input	101101	G_{455} G_{455} G_{453} G_{452} G_{451} G_{450}	78
No Input	101110	G_{465} G_{464} G_{463} G_{462} G_{461} G_{460}	79
No Input	101111	G_{475} G_{474} G_{473} G_{472} G_{471} G_{470}	80
No Input	110000	G_{485} G_{484} G_{483} G_{482} G_{481} G_{480}	81
No Input	110001	G_{495} G_{494} G_{493} G_{492} G_{491} G_{490}	82
No Input	110010	G_{505} G_{504} G_{503} G_{502} G_{501} G_{500}	83
No Input	110011	G_{515} G_{514} G_{513} G_{512} G_{511} G_{510}	84
No Input	110100	G_{525} G_{524} G_{523} G_{522} G_{521} G_{520}	85
No Input	110101	G_{535} G_{534} G_{533} G_{532} G_{531} G_{530}	86
No Input	110110	G_{545} G_{544} G_{543} G_{542} G_{541} G_{540}	87
No Input	110111	G_{555} G_{554} G_{553} G_{552} G_{551} G_{550}	88
No Input	111000	G_{565} G_{564} G_{563} G_{562} G_{561} G_{560}	89
No Input	111001	G_{575} G_{574} G_{573} G_{572} G_{571} G_{570}	90
No Input	111010	G_{585} G_{584} G_{583} G_{582} G_{581} G_{580}	91
No Input	111011	G_{595} G_{594} G_{593} G_{592} G_{591} G_{590}	92
No Input	111100	G_{605} G_{604} G_{603} G_{602} G_{601} G_{600}	93
No Input	111101	G_{615} G_{614} G_{613} G_{612} G_{611} G_{610}	94
No Input	111110	G_{625} G_{624} G_{623} G_{622} G_{621} G_{620}	95
No Input	111111	G_{635} G_{634} G_{633} G_{632} G_{631} G_{630}	96

Table 20 12-bit, 16-bit to 18-bit LUT Blue Component Values

B input (4bit) 12 bit/pixel -mode 4,096 colors	B input (5 bit) 16 bit/pixel -mode 65,536 colors	B output (6bit) 18 bit/pixel -mode 262,144 colors	write_LUT Parameter
0000	00000	$B_{005}\ B_{004}\ B_{003}\ B_{002}\ B_{001}\ B_{000}$	97
0001	00001	$B_{015}\ B_{014}\ B_{013}\ B_{012}\ B_{011}\ B_{010}$	98
0010	00010	$B_{025}\ B_{024}\ B_{023}\ B_{022}\ B_{021}\ B_{020}$	99
0011	00011	$B_{035}\ B_{034}\ B_{033}\ B_{032}\ B_{031}\ B_{030}$	100
0100	00100	$B_{045}\ B_{044}\ B_{043}\ B_{042}\ B_{041}\ B_{040}$	101
0101	00101	$B_{055}\ B_{054}\ B_{053}\ B_{052}\ B_{051}\ B_{050}$	102
0110	00110	$B_{065}\ B_{064}\ B_{063}\ B_{062}\ B_{061}\ B_{060}$	103
0111	00111	$B_{075}\ B_{074}\ B_{073}\ B_{072}\ B_{071}\ B_{070}$	104
1000	01000	$B_{085}\ B_{084}\ B_{083}\ B_{082}\ B_{081}\ B_{080}$	105
1001	01001	$B_{095}\ B_{094}\ B_{093}\ B_{092}\ B_{091}\ B_{090}$	106
1010	01010	$B_{105}\ B_{104}\ B_{103}\ B_{102}\ B_{101}\ B_{100}$	107
1011	01011	$B_{115}\ B_{114}\ B_{113}\ B_{112}\ B_{111}\ B_{110}$	108
1100	01100	$B_{125}\ B_{124}\ B_{123}\ B_{122}\ B_{121}\ B_{120}$	109
1101	01101	$B_{135}\ B_{134}\ B_{133}\ B_{132}\ B_{131}\ B_{130}$	110
1110	01110	$B_{145}\ B_{144}\ B_{143}\ B_{142}\ B_{141}\ B_{140}$	111
1111	01111	$B_{155}\ B_{154}\ B_{153}\ B_{152}\ B_{151}\ B_{150}$	112
No Input	10000	$B_{165}\ B_{164}\ B_{163}\ B_{162}\ B_{161}\ B_{160}$	113
No Input	10001	$B_{175}\ B_{174}\ B_{173}\ B_{172}\ B_{171}\ B_{170}$	114
No Input	10010	$B_{185}\ B_{184}\ B_{183}\ B_{182}\ B_{181}\ B_{180}$	115
No Input	10011	$B_{195}\ B_{194}\ B_{193}\ B_{192}\ B_{191}\ B_{190}$	116
No Input	10100	$B_{205}\ B_{204}\ B_{203}\ B_{202}\ B_{201}\ B_{200}$	117
No Input	10101	$B_{215}\ B_{214}\ B_{213}\ B_{212}\ B_{211}\ B_{210}$	118
No Input	10110	$B_{225}\ B_{224}\ B_{223}\ B_{222}\ B_{221}\ B_{220}$	119
No Input	10111	$B_{235}\ B_{234}\ B_{233}\ B_{232}\ B_{231}\ B_{230}$	120
No Input	11000	$B_{245}\ B_{244}\ B_{243}\ B_{242}\ B_{241}\ B_{240}$	121
No Input	11001	$B_{255}\ B_{254}\ B_{253}\ B_{252}\ B_{251}\ B_{250}$	122
No Input	11010	$B_{265}\ B_{264}\ B_{263}\ B_{262}\ B_{261}\ B_{260}$	123
No Input	11011	$B_{275}\ B_{274}\ B_{273}\ B_{272}\ B_{271}\ B_{270}$	124
No Input	11100	$B_{285}\ B_{284}\ B_{283}\ B_{282}\ B_{281}\ B_{280}$	125
No Input	11101	$B_{295}\ B_{294}\ B_{293}\ B_{292}\ B_{291}\ B_{290}$	126
No Input	11110	$B_{305}\ B_{304}\ B_{303}\ B_{302}\ B_{301}\ B_{300}$	127
No Input	11111	$B_{315}\ B_{314}\ B_{313}\ B_{312}\ B_{311}\ B_{310}$	128

B.3 Color Depth Conversion LUT – 12-bit, 16-bit and 18-bit Colors to 24-bit Color

Table 21 12-bit, 16-bit and 18-bit Colors to 24-bit Color LUT Red Component Values

R input (4bit) 12 bit/pixel - mode 4,096 colors	R input (5 bit) 16 bit/pixel - mode 65,536 colors	R input (6 bit) 18 bit/pixel - mode 262,144 colors	R output (8bit) 24 bit/pixel -mode 16,777,216 colors	write_LUT Parameter
0000	00000	000000	R_{007} R_{006} R_{005} R_{004} R_{003} R_{002} R_{001} R_{000}	1
0001	00001	000001	R_{017} R_{016} R_{015} R_{014} R_{013} R_{012} R_{011} R_{010}	2
0010	00010	000010	R_{027} R_{026} R_{025} R_{024} R_{023} R_{022} R_{021} R_{020}	3
0011	00011	000011	R_{037} R_{036} R_{035} R_{034} R_{033} R_{032} R_{031} R_{030}	4
0100	00100	000100	R_{047} R_{046} R_{045} R_{044} R_{043} R_{042} R_{041} R_{040}	5
0101	00101	000101	R_{057} R_{056} R_{055} R_{054} R_{053} R_{052} R_{051} R_{050}	6
0110	00110	000110	R_{067} R_{066} R_{065} R_{064} R_{063} R_{062} R_{061} R_{060}	7
0111	00111	000111	R_{077} R_{076} R_{075} R_{074} R_{073} R_{072} R_{071} R_{070}	8
1000	01000	001000	R_{087} R_{086} R_{085} R_{084} R_{083} R_{082} R_{081} R_{080}	9
1001	01001	001001	R_{097} R_{096} R_{095} R_{094} R_{093} R_{092} R_{091} R_{090}	10
1010	01010	001010	R_{107} R_{106} R_{105} R_{104} R_{103} R_{102} R_{101} R_{100}	11
1011	01011	001011	R_{117} R_{116} R_{115} R_{114} R_{113} R_{112} R_{111} R_{110}	12
1100	01100	001100	R_{127} R_{126} R_{125} R_{124} R_{123} R_{122} R_{121} R_{120}	13
1101	01101	001101	R_{137} R_{136} R_{135} R_{134} R_{133} R_{132} R_{131} R_{130}	14
1110	01110	001110	R_{147} R_{146} R_{145} R_{144} R_{143} R_{142} R_{141} R_{140}	15
1111	01111	001111	R_{157} R_{156} R_{155} R_{154} R_{153} R_{152} R_{151} R_{150}	16
No Input	10000	010000	R_{167} R_{166} R_{165} R_{164} R_{163} R_{162} R_{161} R_{160}	17
No Input	10001	010001	R_{177} R_{176} R_{175} R_{174} R_{173} R_{172} R_{171} R_{170}	18
No Input	10010	010010	R_{187} R_{186} R_{185} R_{184} R_{183} R_{182} R_{181} R_{180}	19
No Input	10011	010011	R_{197} R_{196} R_{195} R_{194} R_{193} R_{192} R_{191} R_{190}	20
No Input	10100	010100	R_{207} R_{206} R_{205} R_{204} R_{203} R_{202} R_{201} R_{200}	21
No Input	10101	010101	R_{217} R_{216} R_{215} R_{214} R_{213} R_{212} R_{211} R_{210}	22
No Input	10110	010110	R_{227} R_{226} R_{225} R_{224} R_{223} R_{222} R_{221} R_{220}	23
No Input	10111	010111	R_{237} R_{236} R_{235} R_{234} R_{233} R_{232} R_{231} R_{230}	24
No Input	11000	011000	R_{247} R_{246} R_{245} R_{244} R_{243} R_{242} R_{241} R_{240}	25
No Input	11001	011001	R_{257} R_{256} R_{255} R_{254} R_{253} R_{252} R_{251} R_{250}	26
No Input	11010	011010	R_{267} R_{266} R_{265} R_{264} R_{263} R_{262} R_{261} R_{260}	27
No Input	11011	011011	R_{277} R_{276} R_{275} R_{274} R_{273} R_{272} R_{271} R_{270}	28
No Input	11100	011100	R_{287} R_{286} R_{285} R_{284} R_{283} R_{282} R_{281} R_{280}	29
No Input	11101	011101	R_{297} R_{296} R_{295} R_{294} R_{293} R_{292} R_{291} R_{290}	30
No Input	11110	011110	R_{307} R_{306} R_{305} R_{304} R_{303} R_{302} R_{301} R_{300}	31

R input (4bit) **12 bit/pixel - mode** **4,096 colors**	**R input (5 bit)** **16 bit/pixel - mode** **65,536 colors**	**R input (6 bit)** **18 bit/pixel - mode** **262,144 colors**	**R output (8bit)** **24 bit/pixel -mode** **16,777,216 colors**	**write_LUT Parameter**
No Input	11111	011111	R_{317} R_{316} R_{315} R_{314} R_{313} R_{312} R_{311} R_{310}	32
No Input	No Input	100000	R_{327} R_{326} R_{325} R_{324} R_{323} R_{322} R_{321} R_{320}	33
No Input	No Input	100001	R_{337} R_{336} R_{335} R_{334} R_{333} R_{332} R_{331} R_{330}	34
No Input	No Input	100010	R_{347} R_{346} R_{345} R_{344} R_{343} R_{342} R_{341} R_{340}	35
No Input	No Input	100011	R_{357} R_{356} R_{355} R_{354} R_{353} R_{352} R_{351} R_{350}	36
No Input	No Input	100100	R_{367} R_{366} R_{365} R_{364} R_{363} R_{362} R_{361} R_{360}	37
No Input	No Input	100101	R_{377} R_{376} R_{375} R_{374} R_{373} R_{372} R_{371} R_{370}	38
No Input	No Input	100110	R_{387} R_{386} R_{385} R_{384} R_{383} R_{382} R_{381} R_{380}	39
No Input	No Input	100111	R_{397} R_{396} R_{395} R_{394} R_{393} R_{392} R_{391} R_{390}	40
No Input	No Input	101000	R_{407} R_{406} R_{405} R_{404} R_{403} R_{402} R_{401} R_{400}	41
No Input	No Input	101001	R_{417} R_{416} R_{415} R_{414} R_{413} R_{412} R_{411} R_{410}	42
No Input	No Input	101010	R_{427} R_{426} R_{425} R_{424} R_{423} R_{422} R_{421} R_{420}	43
No Input	No Input	101011	R_{437} R_{436} R_{435} R_{434} R_{433} R_{432} R_{431} R_{430}	44
No Input	No Input	101100	R_{447} R_{446} R_{445} R_{444} R_{443} R_{442} R_{441} R_{440}	45
No Input	No Input	101101	R_{457} R_{456} R_{455} R_{454} R_{453} R_{452} R_{451} R_{450}	46
No Input	No Input	101110	R_{467} R_{466} R_{465} R_{464} R_{463} R_{462} R_{461} R_{460}	47
No Input	No Input	101111	R_{477} R_{476} R_{475} R_{474} R_{473} R_{472} R_{471} R_{470}	48
No Input	No Input	110000	R_{487} R_{486} R_{485} R_{484} R_{483} R_{482} R_{481} R_{480}	49
No Input	No Input	110001	R_{497} R_{496} R_{495} R_{494} R_{493} R_{492} R_{491} R_{490}	50
No Input	No Input	110010	R_{507} R_{506} R_{505} R_{504} R_{503} R_{502} R_{501} R_{500}	51
No Input	No Input	110011	R_{517} R_{516} R_{515} R_{514} R_{513} R_{512} R_{511} R_{510}	52
No Input	No Input	110100	R_{527} R_{526} R_{525} R_{524} R_{523} R_{522} R_{521} R_{520}	53
No Input	No Input	110101	R_{537} R_{536} R_{535} R_{534} R_{533} R_{532} R_{531} R_{530}	54
No Input	No Input	110110	R_{547} R_{546} R_{545} R_{544} R_{543} R_{542} R_{541} R_{540}	55
No Input	No Input	110111	R_{557} R_{556} R_{555} R_{554} R_{553} R_{552} R_{551} R_{550}	56
No Input	No Input	111000	R_{567} R_{566} R_{565} R_{564} R_{563} R_{562} R_{561} R_{560}	57
No Input	No Input	111001	R_{577} R_{576} R_{575} R_{574} R_{573} R_{572} R_{571} R_{570}	58
No Input	No Input	111010	R_{587} R_{586} R_{585} R_{584} R_{583} R_{582} R_{581} R_{580}	59
No Input	No Input	111011	R_{597} R_{596} R_{595} R_{594} R_{593} R_{592} R_{591} R_{590}	60
No Input	No Input	111100	R_{607} R_{606} R_{605} R_{604} R_{603} R_{602} R_{601} R_{600}	61
No Input	No Input	111101	R_{617} R_{616} R_{615} R_{614} R_{613} R_{612} R_{611} R_{610}	62
No Input	No Input	111110	R_{627} R_{626} R_{625} R_{624} R_{623} R_{622} R_{621} R_{620}	63
No Input	No Input	111111	R_{637} R_{636} R_{635} R_{634} R_{633} R_{632} R_{631} R_{630}	64

Table 22 12-bit, 16-bit and 18-bit Colors to 24-bit Color LUT Green Component Values

G input (4bit) 12 bit/pixel - mode 4,096 colors	G input (6 bit) 16 bit/pixel - mode 65,536 colors	G input (6 bit) 18 bit/pixel - mode 262,144 colors	G output (8bit) 24 bit/pixel -mode 16,777,216 colors	write_LUT Parameter
0000	000000	000000	$G_{007} G_{006} G_{005} G_{004} G_{003} G_{002} G_{001} G_{000}$	65
0001	000001	000001	$G_{017} G_{016} G_{015} G_{014} G_{013} G_{012} G_{011} G_{010}$	66
0010	000010	000010	$G_{027} G_{026} G_{025} G_{024} G_{023} G_{022} G_{021} G_{020}$	67
0011	000011	000011	$G_{037} G_{036} G_{035} G_{034} G_{033} G_{032} G_{031} G_{030}$	68
0100	000100	000100	$G_{047} G_{046} G_{045} G_{044} G_{043} G_{042} G_{041} G_{040}$	69
0101	000101	000101	$G_{057} G_{056} G_{055} G_{054} G_{053} G_{052} G_{051} G_{050}$	70
0110	000110	000110	$G_{067} G_{066} G_{065} G_{064} G_{063} G_{062} G_{061} G_{060}$	71
0111	000111	000111	$G_{077} G_{076} G_{075} G_{074} G_{073} G_{072} G_{071} G_{070}$	72
1000	001000	001000	$G_{087} G_{086} G_{085} G_{084} G_{083} G_{082} G_{081} G_{080}$	73
1001	001001	001001	$G_{097} G_{096} G_{095} G_{094} G_{093} G_{092} G_{091} G_{090}$	74
1010	001010	001010	$G_{107} G_{106} G_{105} G_{104} G_{103} G_{102} G_{101} G_{100}$	75
1011	001011	001011	$G_{117} G_{116} G_{115} G_{114} G_{113} G_{112} G_{111} G_{110}$	76
1100	001100	001100	$G_{127} G_{126} G_{125} G_{124} G_{123} G_{122} G_{121} G_{120}$	77
1101	001101	001101	$G_{137} G_{136} G_{135} G_{134} G_{133} G_{132} G_{131} G_{130}$	78
1110	001110	001110	$G_{147} G_{146} G_{145} G_{144} G_{143} G_{142} G_{141} G_{140}$	79
1111	001111	001111	$G_{157} G_{156} G_{155} G_{154} G_{153} G_{152} G_{151} G_{150}$	80
No Input	010000	010000	$G_{167} G_{166} G_{165} G_{164} G_{163} G_{162} G_{161} G_{160}$	81
No Input	010001	010001	$G_{177} G_{176} G_{175} G_{174} G_{173} G_{172} G_{171} G_{170}$	82
No Input	010010	010010	$G_{187} G_{186} G_{185} G_{184} G_{183} G_{182} G_{181} G_{180}$	83
No Input	010011	010011	$G_{197} G_{196} G_{195} G_{194} G_{193} G_{192} G_{191} G_{190}$	84
No Input	010100	010100	$G_{207} G_{206} G_{205} G_{204} G_{203} G_{202} G_{201} G_{200}$	85
No Input	010101	010101	$G_{217} G_{216} G_{215} G_{214} G_{213} G_{212} G_{211} G_{210}$	86
No Input	010110	010110	$G_{227} G_{226} G_{225} G_{224} G_{223} G_{222} G_{221} G_{220}$	87
No Input	010111	010111	$G_{237} G_{236} G_{235} G_{234} G_{233} G_{232} G_{231} G_{230}$	88
No Input	011000	011000	$G_{247} G_{246} G_{245} G_{244} G_{243} G_{242} G_{241} G_{240}$	89
No Input	011001	011001	$G_{257} G_{256} G_{255} G_{254} G_{253} G_{252} G_{251} G_{250}$	90
No Input	011010	011010	$G_{267} G_{266} G_{265} G_{264} G_{263} G_{262} G_{261} G_{260}$	91
No Input	011011	011011	$G_{277} G_{276} G_{275} G_{274} G_{273} G_{272} G_{271} G_{270}$	92
No Input	011100	011100	$G_{287} G_{286} G_{285} G_{284} G_{283} G_{282} G_{281} G_{280}$	93
No Input	011101	011101	$G_{297} G_{296} G_{295} G_{294} G_{293} G_{292} G_{291} G_{290}$	94
No Input	011110	011110	$G_{307} G_{306} G_{305} G_{304} G_{303} G_{302} G_{301} G_{300}$	95
No Input	011111	011111	$G_{317} G_{316} G_{315} G_{314} G_{313} G_{312} G_{311} G_{310}$	96
No Input	100000	100000	$G_{327} G_{326} G_{325} G_{324} G_{323} G_{322} G_{321} G_{320}$	97

G input (4bit) **12 bit/pixel - mode** **4,096 colors**	**G input (6 bit)** **16 bit/pixel - mode** **65,536 colors**	**G input (6 bit)** **18 bit/pixel - mode** **262,144 colors**	**G output (8bit)** **24 bit/pixel -mode** **16,777,216 colors**	**write_LUT Parameter**
No Input	100001	100001	G_{337} G_{336} G_{335} G_{334} G_{333} G_{332} G_{331} G_{330}	98
No Input	100010	100010	G_{347} G_{346} G_{345} G_{344} G_{343} G_{342} G_{341} G_{340}	99
No Input	100011	100011	G_{357} G_{356} G_{355} G_{354} G_{353} G_{352} G_{351} G_{350}	100
No Input	100100	100100	G_{367} G_{366} G_{365} G_{364} G_{363} G_{362} G_{361} G_{360}	101
No Input	100101	100101	G_{377} G_{376} G_{375} G_{374} G_{373} G_{372} G_{371} G_{370}	102
No Input	100110	100110	G_{387} G_{386} G_{385} G_{384} G_{383} G_{382} G_{381} G_{380}	103
No Input	100111	100111	G_{397} G_{396} G_{395} G_{394} G_{393} G_{392} G_{391} G_{390}	104
No Input	101000	101000	G_{407} G_{406} G_{405} G_{404} G_{403} G_{402} G_{401} G_{400}	105
No Input	101001	101001	G_{417} G_{416} G_{415} G_{414} G_{413} G_{412} G_{411} G_{410}	106
No Input	101010	101010	G_{427} G_{426} G_{425} G_{424} G_{423} G_{422} G_{421} G_{420}	107
No Input	101011	101011	G_{437} G_{436} G_{435} G_{434} G_{433} G_{432} G_{431} G_{430}	108
No Input	101100	101100	G_{447} G_{446} G_{445} G_{444} G_{443} G_{442} G_{441} G_{440}	109
No Input	101101	101101	G_{457} G_{456} G_{455} G_{454} G_{453} G_{452} G_{451} G_{450}	110
No Input	101110	101110	G_{467} G_{466} G_{465} G_{464} G_{463} G_{462} G_{461} G_{460}	111
No Input	101111	101111	G_{477} G_{476} G_{475} G_{474} G_{473} G_{472} G_{471} G_{470}	112
No Input	110000	110000	G_{487} G_{486} G_{485} G_{484} G_{483} G_{482} G_{481} G_{480}	113
No Input	110001	110001	G_{497} G_{496} G_{495} G_{494} G_{493} G_{492} G_{491} G_{490}	114
No Input	110010	110010	G_{507} G_{506} G_{505} G_{504} G_{503} G_{502} G_{501} G_{500}	115
No Input	110011	110011	G_{517} G_{516} G_{515} G_{514} G_{513} G_{512} G_{511} G_{510}	116
No Input	110100	110100	G_{527} G_{526} G_{525} G_{524} G_{523} G_{522} G_{521} G_{520}	117
No Input	110101	110101	G_{537} G_{536} G_{535} G_{534} G_{533} G_{532} G_{531} G_{530}	118
No Input	110110	110110	G_{547} G_{546} G_{545} G_{544} G_{543} G_{542} G_{541} G_{540}	119
No Input	110111	110111	G_{557} G_{556} G_{555} G_{554} G_{553} G_{552} G_{551} G_{550}	120
No Input	111000	111000	G_{567} G_{566} G_{565} G_{564} G_{563} G_{562} G_{561} G_{560}	121
No Input	111001	111001	G_{577} G_{576} G_{575} G_{574} G_{573} G_{572} G_{571} G_{570}	122
No Input	111010	111010	G_{587} G_{586} G_{585} G_{584} G_{583} G_{582} G_{581} G_{580}	123
No Input	111011	111011	G_{597} G_{596} G_{595} G_{594} G_{593} G_{592} G_{591} G_{590}	124
No Input	111100	111100	G_{607} G_{606} G_{605} G_{604} G_{603} G_{602} G_{601} G_{600}	125
No Input	111101	111101	G_{617} G_{616} G_{615} G_{614} G_{613} G_{612} G_{611} G_{610}	126
No Input	111110	111110	G_{627} G_{626} G_{625} G_{624} G_{623} G_{622} G_{621} G_{620}	127
No Input	111111	111111	G_{637} G_{636} G_{635} G_{634} G_{633} G_{632} G_{631} G_{630}	128

Table 23 12-bit, 16-bit and 18-bit Colors to 24-bit Color LUT Blue Component Values

B input (4bit) **12 bit/pixel - mode** **4,096 colors**	**B input (5 bit)** **16 bit/pixel - mode** **65,536 colors**	**B input (6 bit)** **18 bit/pixel - mode** **262,144 colors**	**B output (8bit)** **24 bit/pixel -mode** **16,777,216 colors**	**write_LUT Parameter**
0000	00000	000000	$B_{007}\ B_{006}\ B_{005}\ B_{004}\ B_{003}\ B_{002}\ B_{001}\ B_{000}$	129
0001	00001	000001	$B_{017}\ B_{016}\ B_{015}\ B_{014}\ B_{013}\ B_{012}\ B_{011}\ B_{010}$	130
0010	00010	000010	$B_{027}\ B_{026}\ B_{025}\ B_{024}\ B_{023}\ B_{022}\ B_{021}\ B_{020}$	131
0011	00011	000011	$B_{037}\ B_{036}\ B_{035}\ B_{034}\ B_{033}\ B_{032}\ B_{031}\ B_{030}$	132
0100	00100	000100	$B_{047}\ B_{046}\ B_{045}\ B_{044}\ B_{043}\ B_{042}\ B_{041}\ B_{040}$	133
0101	00101	000101	$B_{057}\ B_{056}\ B_{055}\ B_{054}\ B_{053}\ B_{052}\ B_{051}\ B_{050}$	134
0110	00110	000110	$B_{067}\ B_{066}\ B_{065}\ B_{064}\ B_{063}\ B_{062}\ B_{061}\ B_{060}$	135
0111	00111	000111	$B_{077}\ B_{076}\ B_{075}\ B_{074}\ B_{073}\ B_{072}\ B_{071}\ B_{070}$	136
1000	01000	001000	$B_{087}\ B_{086}\ B_{085}\ B_{084}\ B_{083}\ B_{082}\ B_{081}\ B_{080}$	137
1001	01001	001001	$B_{097}\ B_{096}\ B_{095}\ B_{094}\ B_{093}\ B_{092}\ B_{091}\ B_{090}$	138
1010	01010	001010	$B_{107}\ B_{106}\ B_{105}\ B_{104}\ B_{103}\ B_{102}\ B_{101}\ B_{100}$	139
1011	01011	001011	$B_{117}\ B_{116}\ B_{115}\ B_{114}\ B_{113}\ B_{112}\ B_{111}\ B_{110}$	140
1100	01100	001100	$B_{127}\ B_{126}\ B_{125}\ B_{124}\ B_{123}\ B_{122}\ B_{121}\ B_{120}$	141
1101	01101	001101	$B_{137}\ B_{136}\ B_{135}\ B_{134}\ B_{133}\ B_{132}\ B_{131}\ B_{130}$	142
1110	01110	001110	$B_{147}\ B_{146}\ B_{145}\ B_{144}\ B_{143}\ B_{142}\ B_{141}\ B_{140}$	143
1111	01111	001111	$B_{157}\ B_{156}\ B_{155}\ B_{154}\ B_{153}\ B_{152}\ B_{151}\ B_{150}$	144
No Input	10000	010000	$B_{167}\ B_{166}\ B_{165}\ B_{164}\ B_{163}\ B_{162}\ B_{161}\ B_{160}$	145
No Input	10001	010001	$B_{177}\ B_{176}\ B_{175}\ B_{174}\ B_{173}\ B_{172}\ B_{171}\ B_{170}$	146
No Input	10010	010010	$B_{187}\ B_{186}\ B_{185}\ B_{184}\ B_{183}\ B_{182}\ B_{181}\ B_{180}$	147
No Input	10011	010011	$B_{197}\ B_{196}\ B_{195}\ B_{194}\ B_{193}\ B_{192}\ B_{191}\ B_{190}$	148
No Input	10100	010100	$B_{207}\ B_{206}\ B_{205}\ B_{204}\ B_{203}\ B_{202}\ B_{201}\ B_{200}$	149
No Input	10101	010101	$B_{217}\ B_{216}\ B_{215}\ B_{214}\ B_{213}\ B_{212}\ B_{211}\ B_{210}$	150
No Input	10110	010110	$B_{227}\ B_{226}\ B_{225}\ B_{224}\ B_{223}\ B_{222}\ B_{221}\ B_{220}$	151
No Input	10111	010111	$B_{237}\ B_{236}\ B_{235}\ B_{234}\ B_{233}\ B_{232}\ B_{231}\ B_{230}$	152
No Input	11000	011000	$B_{247}\ B_{246}\ B_{245}\ B_{244}\ B_{243}\ B_{242}\ B_{241}\ B_{240}$	153
No Input	11001	011001	$B_{257}\ B_{256}\ B_{255}\ B_{254}\ B_{253}\ B_{252}\ B_{251}\ B_{250}$	154
No Input	11010	011010	$B_{267}\ B_{266}\ B_{265}\ B_{264}\ B_{263}\ B_{262}\ B_{261}\ B_{260}$	155
No Input	11011	011011	$B_{277}\ B_{276}\ B_{275}\ B_{274}\ B_{273}\ B_{272}\ B_{271}\ B_{270}$	156
No Input	11100	011100	$B_{287}\ B_{286}\ B_{285}\ B_{284}\ B_{283}\ B_{282}\ B_{281}\ B_{280}$	157
No Input	11101	011101	$B_{297}\ B_{296}\ B_{295}\ B_{294}\ B_{293}\ B_{292}\ B_{291}\ B_{290}$	158
No Input	11110	011110	$B_{307}\ B_{306}\ B_{305}\ B_{304}\ B_{303}\ B_{302}\ B_{301}\ B_{300}$	159
No Input	11111	011111	$B_{317}\ B_{316}\ B_{315}\ B_{314}\ B_{313}\ B_{312}\ B_{311}\ B_{310}$	160
No Input	No Input	100000	$B_{327}\ B_{326}\ B_{325}\ B_{324}\ B_{323}\ B_{322}\ B_{321}\ B_{320}$	161

B input (4bit) **12 bit/pixel - mode** **4,096 colors**	**B input (5 bit)** **16 bit/pixel - mode** **65,536 colors**	**B input (6 bit)** **18 bit/pixel - mode** **262,144 colors**	**B output (8bit)** **24 bit/pixel -mode** **16,777,216 colors**	**write_LUT Parameter**
No Input	No Input	100001	B_{337} B_{336} B_{335} B_{334} B_{333} B_{332} B_{331} B_{330}	162
No Input	No Input	100010	B_{347} B_{346} B_{345} B_{344} B_{343} B_{342} B_{341} B_{340}	163
No Input	No Input	100011	B_{357} B_{356} B_{355} B_{354} B_{353} B_{352} B_{351} B_{350}	164
No Input	No Input	100100	B_{367} B_{366} B_{365} B_{364} B_{363} B_{362} B_{361} B_{360}	165
No Input	No Input	100101	B_{377} B_{376} B_{375} B_{374} B_{373} B_{372} B_{371} B_{370}	166
No Input	No Input	100110	B_{387} B_{386} B_{385} B_{384} B_{383} B_{382} B_{381} B_{380}	167
No Input	No Input	100111	B_{397} B_{396} B_{395} B_{394} B_{393} B_{392} B_{391} B_{390}	168
No Input	No Input	101000	B_{407} B_{406} B_{405} B_{404} B_{403} B_{402} B_{401} B_{400}	169
No Input	No Input	101001	B_{417} B_{416} B_{415} B_{414} B_{413} B_{412} B_{411} B_{410}	170
No Input	No Input	101010	B_{427} B_{426} B_{425} B_{424} B_{423} B_{422} B_{421} B_{420}	171
No Input	No Input	101011	B_{437} B_{436} B_{435} B_{434} B_{433} B_{432} B_{431} B_{430}	172
No Input	No Input	101100	B_{447} B_{446} B_{445} B_{444} B_{443} B_{442} B_{441} B_{440}	173
No Input	No Input	101101	B_{457} B_{456} B_{455} B_{454} B_{453} B_{452} B_{451} B_{450}	174
No Input	No Input	101110	B_{467} B_{466} B_{465} B_{464} B_{463} B_{462} B_{461} B_{460}	175
No Input	No Input	101111	B_{477} B_{476} B_{475} B_{474} B_{473} B_{472} B_{471} B_{470}	176
No Input	No Input	110000	B_{487} B_{486} B_{485} B_{484} B_{483} B_{482} B_{481} B_{480}	177
No Input	No Input	110001	B_{497} B_{496} B_{495} B_{494} B_{493} B_{492} B_{491} B_{490}	178
No Input	No Input	110010	B_{507} B_{506} B_{505} B_{504} B_{503} B_{502} B_{501} B_{500}	179
No Input	No Input	110011	B_{517} B_{516} B_{515} B_{514} B_{513} B_{512} B_{511} B_{510}	180
No Input	No Input	110100	B_{527} B_{526} B_{525} B_{524} B_{523} B_{522} B_{521} B_{520}	181
No Input	No Input	110101	B_{537} B_{536} B_{535} B_{534} B_{533} B_{532} B_{531} B_{530}	182
No Input	No Input	110110	B_{547} B_{546} B_{545} B_{544} B_{543} B_{542} B_{541} B_{540}	183
No Input	No Input	110111	B_{557} B_{556} B_{555} B_{554} B_{553} B_{552} B_{551} B_{550}	184
No Input	No Input	111000	B_{567} B_{566} B_{565} B_{564} B_{563} B_{562} B_{561} B_{560}	185
No Input	No Input	111001	B_{577} B_{576} B_{575} B_{574} B_{573} B_{572} B_{571} B_{570}	186
No Input	No Input	111010	B_{587} B_{586} B_{585} B_{584} B_{583} B_{582} B_{581} B_{580}	187
No Input	No Input	111011	B_{597} B_{596} B_{595} B_{594} B_{593} B_{592} B_{591} B_{590}	188
No Input	No Input	111100	B_{607} B_{606} B_{605} B_{604} B_{603} B_{602} B_{601} B_{600}	189
No Input	No Input	111101	B_{617} B_{616} B_{615} B_{614} B_{613} B_{612} B_{611} B_{610}	190
No Input	No Input	111110	B_{627} B_{626} B_{625} B_{624} B_{623} B_{622} B_{621} B_{620}	191
No Input	No Input	111111	B_{637} B_{636} B_{635} B_{634} B_{633} B_{632} B_{631} B_{630}	192

Participants

The following list includes those persons who participated in the Working Group that developed this specification and who consented to appear on this list.

Nausheen Ansari, Intel Corporation

Quinn Carter, ARM Limited

Edo Cohen, Marvell

Stephen Creaney, Cadence Design Systems, Inc.

Amir Dafnai, Cadence Design Systems, Inc.

Rob Frizzell, Atmel Corporation

Karan Galhotra, Synopsys, Inc.

James Goel, Qualcomm Incorporated

Wei Han, Lattice Semiconductor Corp.

Jim Hunkins, Advanced Micro Devices, Inc.

Samson Kim, Qualcomm Incorporated

Tim SangKyu Lee, Samsung Electronics, Co.

George Letey, Microsoft Corporation

Ryan Liu, Advanced Micro Devices, Inc.

Thirumal Molagounder, GDA Technologies

Kit Fong Ng, Qualcomm Incorporated

Pablo Ortega, NVIDIA

Josh Pan, MediaTek Inc.

Juha Pankala, Microsoft Corporation

Sungjin Park, LG Electronics, Inc.

Alex Passi, Cadence Design Systems, Inc.

Richard Petrie, DisplayLink (UK) Ltd.

Jayavarapu Rao, ARM Limited

James Rippie, MIPI Alliance

Hugo Santos, Synopsys, Inc.

Neeraj Sharma, Synopsys, Inc.

Jeffrey Small, Synaptics

Dale Stolitzka, Samsung Electronics, Co.

Nobu Suzuki, Intel Corporation

Seshi Veerapally, NVIDIA

Rick Wietfeldt, Qualcomm Incorporated

George Wiley, Qualcomm Incorporated

Allen Sooyoung Woo, Samsung Electronics, Co.

Jie Zhou, Advanced Micro Devices, Inc.

Larkin Zhang, Advanced Micro Devices, Inc.