



MIPI Alliance Specification for Display Command Set

Version 1.02.00 – 23 July 2009

MIPI Board Approved 24-Nov-2009

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

1 NOTICE OF DISCLAIMER

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

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

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

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

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

36 MIPI Alliance, Inc.
37 c/o IEEE-ISTO
38 445 Hoes Lane
39 Piscataway, NJ 08854
40 Attn: Board Secretary

41

42 Contents

| | | |
|----|--|----|
| 43 | Version 1.02.00 – 23 July 2009..... | i |
| 44 | 1 Overview | 11 |
| 45 | 1.1 Scope | 11 |
| 46 | 1.2 Purpose | 11 |
| 47 | 2 Document Terminology..... | 12 |
| 48 | 2.1 Glossary..... | 12 |
| 49 | 2.2 Acronyms and Abbreviations | 13 |
| 50 | 3 References | 14 |
| 51 | 4 Display Architectures | 15 |
| 52 | 5 Display Functional Description | 18 |
| 53 | 5.1 Power Level Definition | 18 |
| 54 | 5.2 Gamma Curves | 21 |
| 55 | 5.2.1 Gamma Curve 1 (GC0)..... | 21 |
| 56 | 5.2.2 Gamma Curve 2 (GC1)..... | 21 |
| 57 | 5.2.3 Gamma Curve 3 (GC2)..... | 22 |
| 58 | 5.2.4 Gamma Curve 4 (GC3)..... | 22 |
| 59 | 5.3 Self-diagnostic Functions | 23 |
| 60 | 5.3.1 Register Loading Detection | 23 |
| 61 | 5.3.2 Functionality Detection | 24 |
| 62 | 5.3.3 Chip Attachment Detection (optional)..... | 25 |
| 63 | 5.3.4 Display Glass Break Detection (optional) | 26 |
| 64 | 5.4 Display Command Set..... | 27 |
| 65 | 5.5 Command List | 28 |
| 66 | 5.6 Command Accessibility..... | 31 |
| 67 | 5.7 Default Modes and Values | 33 |
| 68 | 6 Command Description..... | 37 |

| | | | |
|----|------|-----------------------------|----|
| 69 | 6.1 | enter_idle_mode | 38 |
| 70 | 6.2 | enter_invert_mode | 40 |
| 71 | 6.3 | enter_normal_mode | 41 |
| 72 | 6.4 | enter_partial_mode | 42 |
| 73 | 6.5 | enter_sleep_mode | 43 |
| 74 | 6.6 | exit_idle_mode | 45 |
| 75 | 6.7 | exit_invert_mode | 46 |
| 76 | 6.8 | exit_sleep_mode | 47 |
| 77 | 6.9 | get_address_mode | 49 |
| 78 | 6.10 | get_blue_channel | 51 |
| 79 | 6.11 | get_diagnostic_result | 52 |
| 80 | 6.12 | get_display_mode | 53 |
| 81 | 6.13 | get_green_channel | 55 |
| 82 | 6.14 | get_pixel_format | 56 |
| 83 | 6.15 | get_power_mode | 58 |
| 84 | 6.16 | get_red_channel | 60 |
| 85 | 6.17 | get_scanline | 61 |
| 86 | 6.18 | get_signal_mode | 62 |
| 87 | 6.19 | nop | 63 |
| 88 | 6.20 | read_DDB_continue | 64 |
| 89 | 6.21 | read_DDB_start | 65 |
| 90 | 6.22 | read_memory_continue | 67 |
| 91 | 6.23 | read_memory_start | 69 |
| 92 | 6.24 | set_address_mode | 71 |
| 93 | 6.25 | set_column_address | 76 |
| 94 | 6.26 | set_display_off | 78 |
| 95 | 6.27 | set_display_on | 79 |
| 96 | 6.28 | set_gamma_curve | 80 |

| | | | |
|-----|---|--|-----|
| 97 | 6.29 | set_page_address | 81 |
| 98 | 6.30 | set_partial_columns | 83 |
| 99 | 6.31 | set_partial_rows..... | 88 |
| 100 | 6.32 | set_pixel_format | 90 |
| 101 | 6.33 | set_scroll_area | 91 |
| 102 | 6.34 | set_scroll_start..... | 94 |
| 103 | 6.35 | set_tear_off..... | 96 |
| 104 | 6.36 | set_tear_on..... | 97 |
| 105 | 6.37 | set_tear_scanline..... | 99 |
| 106 | 6.38 | soft_reset | 101 |
| 107 | 6.39 | write_LUT | 102 |
| 108 | 6.40 | write_memory_continue | 104 |
| 109 | 6.41 | write_memory_start..... | 106 |
| 110 | Annex A Pixel-to-Byte Mapping..... | | 108 |
| 111 | A.1 | Three Bits per Pixel Format..... | 108 |
| 112 | A.2 | Eight Bits per Pixel Format | 109 |
| 113 | A.3 | Twelve Bits per Pixel Format | 109 |
| 114 | A.4 | Sixteen Bits per Pixel Format | 110 |
| 115 | A.5 | Eighteen Bits per Pixel Format..... | 110 |
| 116 | A.6 | Twenty-four Bits per Pixel Format..... | 111 |
| 117 | Annex B Color Depth Conversion Look-up Tables (informative) | | 112 |
| 118 | B.1 | Color Depth Conversion LUT – 12-bit Color to 16-bit Color | 112 |
| 119 | B.2 | Color Depth Conversion LUT – 12-bit and 16-bit Colors to 18-bit Color | 115 |
| 120 | B.3 | Color Depth Conversion LUT – 12-bit, 16-bit and 18-bit Colors to 24-bit Color..... | 120 |
| 121 | | | |
| 122 | | | |

| | | |
|-----|--|----|
| 123 | Figures | |
| 124 | Figure 1 Type 1 Display Architecture Block Diagram..... | 15 |
| 125 | Figure 2 Type 2 Display Architecture Block Diagram..... | 16 |
| 126 | Figure 3 Type 3 Display Architecture Block Diagram..... | 17 |
| 127 | Figure 4 Type 1 Display Architecture Power Change Sequences..... | 19 |
| 128 | Figure 5 Type 2 Display Architecture Power Change Sequence..... | 20 |
| 129 | Figure 6 Type 3 Display Architecture Power Change Sequence..... | 20 |
| 130 | Figure 7 Gamma curve 1 (GC0)..... | 21 |
| 131 | Figure 8 Gamma Curve 2 (GC1)..... | 21 |
| 132 | Figure 9 Gamma Curve 3 (GC2)..... | 22 |
| 133 | Figure 10 Gamma Curve 4 (GC3)..... | 22 |
| 134 | Figure 11 Register Loading Detection Flow Chart..... | 23 |
| 135 | Figure 12 Functionality Detection Flow Chart..... | 24 |
| 136 | Figure 13 Chip Attachment Detection Reference..... | 25 |
| 137 | Figure 14 Chip Attachment Detection Flow Chart..... | 25 |
| 138 | Figure 15 Display Glass Break Detection Reference..... | 26 |
| 139 | Figure 16 Display Glass Break Detection Flow Chart..... | 26 |
| 140 | Figure 17 Flowchart Legend..... | 37 |
| 141 | Figure 18 enter_idle_mode Example..... | 38 |
| 142 | Figure 19 enter_idle_mode Flow Chart..... | 39 |
| 143 | Figure 20 enter_invert_mode Example..... | 40 |
| 144 | Figure 21 enter_invert_mode Flow Chart..... | 40 |
| 145 | Figure 22 enter_sleep_mode Flow Chart..... | 44 |
| 146 | Figure 23 exit_idle_mode Flow Chart..... | 45 |
| 147 | Figure 24 exit_invert_mode Example..... | 46 |
| 148 | Figure 25 exit_invert_mode Flow Chart..... | 46 |
| 149 | Figure 26 exit_sleep_mode Flow Chart..... | 48 |

| | | |
|-----|---|----|
| 150 | Figure 27 get_address_mode Flow Chart | 50 |
| 151 | Figure 28 get_blue_channel Flow Chart..... | 51 |
| 152 | Figure 29 get_diagnostic_result Flow Chart..... | 52 |
| 153 | Figure 30 get_display_mode Flow Chart | 54 |
| 154 | Figure 31 get_green_channel Flow Chart..... | 55 |
| 155 | Figure 32 get_pixel_format Flow Chart | 57 |
| 156 | Figure 33 get_power_mode Flow Chart..... | 59 |
| 157 | Figure 34 get_red_channel Flow Chart | 60 |
| 158 | Figure 35 get_scanline Flow Chart..... | 61 |
| 159 | Figure 36 get_signal_mode Flow Chart | 62 |
| 160 | Figure 37 read_DDB_continue Flow Chart..... | 64 |
| 161 | Figure 38 read_DDB_start Flow Chart..... | 66 |
| 162 | Figure 39 read_memory_continue Flow Chart..... | 68 |
| 163 | Figure 40 read_memory_start Flow Chart..... | 70 |
| 164 | Figure 41 B7 Page Address Order | 71 |
| 165 | Figure 42 B6 Column Address Order | 72 |
| 166 | Figure 43 B5 Page/Column Addressing Order | 72 |
| 167 | Figure 44 B3 RGB Order | 73 |
| 168 | Figure 45 B1 Flip Horizontal..... | 74 |
| 169 | Figure 46 B0 Flip Vertical..... | 74 |
| 170 | Figure 47 set_address_mode Flow Chart | 75 |
| 171 | Figure 48 set_column_address Example | 76 |
| 172 | Figure 49 set_column_address Flow Chart | 77 |
| 173 | Figure 50 set_display_off Example..... | 78 |
| 174 | Figure 51 set_display_off Flow Chart | 78 |
| 175 | Figure 52 set_display_on Example..... | 79 |
| 176 | Figure 53 set_display_on Flow Chart..... | 79 |
| 177 | Figure 54 set_gamma_curve Flow Chart..... | 80 |

| | | |
|-----|---|-----|
| 178 | Figure 55 set_page_address Example..... | 81 |
| 179 | Figure 56 set_page_address Flow Chart..... | 82 |
| 180 | Figure 57 set_partial_columns with set_address_mode B2 = 0..... | 84 |
| 181 | Figure 58 set_partial_columns with set_address_mode B2=1..... | 84 |
| 182 | Figure 59 set_partial_columns with set_address_mode B2 = 0..... | 85 |
| 183 | Figure 60 set_partial_columns with set_address_mode B2 = 1..... | 85 |
| 184 | Figure 61 Entering Partial Display Mode Flow Chart..... | 86 |
| 185 | Figure 62 Exiting Partial Display Mode Flow Chart..... | 87 |
| 186 | Figure 63 set_partial_rows with set_address_mode B4 = 0..... | 88 |
| 187 | Figure 64 set_partial_rows with set_address_mode B4=1..... | 89 |
| 188 | Figure 65 set_partial_rows with set_address_mode B4 = 0..... | 89 |
| 189 | Figure 66 set_partial_rows with set_address_mode B4 = 1..... | 89 |
| 190 | Figure 67 set_pixel_format Flow Chart..... | 90 |
| 191 | Figure 68 set_scroll_area set_address_mode B4 = 1 Example..... | 92 |
| 192 | Figure 69 set_scroll_area set_address_mode B4 = 1 Example..... | 92 |
| 193 | Figure 70 set_scroll_area Flow Chart..... | 93 |
| 194 | Figure 71 set_scroll_start set_address_mode B4 = 0..... | 94 |
| 195 | Figure 72 set_scroll_start set_address_mode B4 = 1..... | 95 |
| 196 | Figure 73 set_tear_off Flow Chart..... | 96 |
| 197 | Figure 74 set_tear_on M = 0..... | 97 |
| 198 | Figure 75 set_tear_on M = 1..... | 97 |
| 199 | Figure 76 set_tear_on Flow Chart..... | 98 |
| 200 | Figure 77 set_tear_scanline..... | 99 |
| 201 | Figure 78 set_tear_scanline Flow Chart..... | 100 |
| 202 | Figure 79 soft_reset Flow Chart..... | 101 |
| 203 | Figure 80 write_LUT Flow Chart..... | 103 |
| 204 | Figure 81 write_memory_continue Flow Chart..... | 105 |
| 205 | Figure 82 write_memory_start Flow Chart..... | 107 |

206 Figure 83 Three Bits per Pixel Format to Byte Mapping108

207 Figure 84 Eight Bits per Pixel Format to Byte Mapping109

208 Figure 85 Twelve Bits per Pixel Format to Byte Mapping.....109

209 Figure 86 Sixteen Bits per Pixel Format to Byte Mapping.....110

210 Figure 87 Eighteen Bits per Pixel Format to Byte Mapping110

211 Figure 88 Twenty-four Bits per Pixel Format to Byte Mapping111

212

213

214 **Tables**

| | | |
|-----|---|-----|
| 215 | Table 1 Command List | 28 |
| 216 | Table 2 Command Accessibility..... | 31 |
| 217 | Table 3 Default Display Mode, Power Mode and Register Values | 33 |
| 218 | Table 4 enter_idle_mode Memory Content vs. Display Color | 38 |
| 219 | Table 5 Gamma Curve Selection..... | 53 |
| 220 | Table 6 Interface Pixel Formats..... | 56 |
| 221 | Table 7 Gamma Curves | 80 |
| 222 | Table 8 LUT Color Depth Conversions..... | 103 |
| 223 | Table 9 Common Color Encoding | 105 |
| 224 | Table 10 Common Color Encoding..... | 107 |
| 225 | Table 11 12-bit to 16-bit LUT Red Component Values | 112 |
| 226 | Table 12 12-bit to 16-bit LUT Green Component Values..... | 113 |
| 227 | Table 13 12-bit to 16-bit LUT Blue Component Values | 114 |
| 228 | Table 14 12-bit, 16-bit to 18-bit LUT Red Component Values..... | 115 |
| 229 | Table 15 12-bit, 16-bit to 18-bit LUT Green Component Values | 117 |
| 230 | Table 16 12-bit, 16-bit to 18-bit LUT Blue Component Values..... | 119 |
| 231 | Table 17 12-bit, 16-bit and 18-bit Colors to 24-bit Color LUT Red Component Values..... | 120 |
| 232 | Table 18 12-bit, 16-bit and 18-bit Colors to 24-bit Color LUT Green Component Values..... | 123 |
| 233 | Table 19 12-bit, 16-bit and 18-bit Colors to 24-bit Color LUT Blue Component Values..... | 126 |
| 234 | | |

235 **MIPI Alliance Specification for Display Command** 236 **Set**

237 **1 Overview**

238 The Display Command Set specification defines display module behavior for devices that adhere to the
239 MIPI specifications for mobile device host processor, and display interfaces in an abstract, device
240 independent way. All commands in this specification shall be supported by display modules that adhere to
241 *MIPI Alliance Standard for Display Pixel Interface* [MIPI01], *MIPI Alliance Standard for Display Bus*
242 *Interface* [MIPI02], and *MIPI Alliance Specification for Display Serial Interface* [MIPI03] except as
243 provided for in the individual specifications.

244 **1.1 Scope**

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

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

248 **1.2 Purpose**

249 The Display Command Set specification is used by manufacturers to design products that adhere to MIPI
250 specifications for mobile device host processor and display interfaces.

251 Implementing the DCS specification reduces the time-to-market and design cost of mobile devices by
252 simplifying the interconnection of products from different manufacturers. In addition, adding new features
253 such as larger or additional displays to mobile devices is simplified due to the extensible nature of the MIPI
254 specifications.
255

256 2 Document Terminology

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

259 The word *shall* is used to indicate mandatory requirements strictly to be followed in order
260 to conform to the standard and from which no deviation is permitted (*shall equals is*
261 *required to*).

262 The use of the word *must* is deprecated and shall not be used when stating mandatory
263 requirements; *must* is used only to describe unavoidable situations.

264 The use of the word *will* is deprecated and shall not be used when stating mandatory
265 requirements; *will* is only used in statements of fact.

266 The word *should* is used to indicate that among several possibilities one is recommended
267 as particularly suitable, without mentioning or excluding others; or that a certain course
268 of action is preferred but not necessarily required; or that (in the negative form) a certain
269 course of action is deprecated but not prohibited (*should equals is recommended that*).

270 The word *may* is used to indicate a course of action permissible within the limits of the
271 standard (*may equals is permitted*).

272 The word *can* is used for statements of possibility and capability, whether material,
273 physical, or causal (*can equals is able to*).

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

275 2.1 Glossary

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

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

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

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

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

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

285 **Display Panel:** Same as Display Device.

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

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

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

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

297 **2.2 Acronyms and Abbreviations**

298 The following acronyms and abbreviations are used throughout this document:

| | | |
|-----|------------|--------------------------|
| 299 | DBI | Display Bus Interface |
| 300 | DCS | Display Command Set |
| 301 | DPI | Display Pixel Interface |
| 302 | DSI | Display Serial Interface |
| 303 | | |

304 **3 References**

- 305 [MIP101] *MIPI Alliance Standard for Display Pixel Interface (DPI-2)*, version 2.00, MIPI Alliance,
306 Inc., 15 September 2005
- 307 [MIP102] *MIPI Alliance Standard for Display Bus Interface (DBI-2)*, version 2.00, MIPI Alliance,
308 Inc., 29 November 2005
- 309 [MIP103] *MIPI Alliance Specification for Display Serial Interface (DSI)*, version 1.02.00, MIPI
310 Alliance, Inc., In Press
- 311 [MIP104] *MIPI Alliance Specification for Device Descriptor Block (DDB)*, version 0.82.01, MIPI
312 Alliance, Inc., 29 October 2008
- 313
- 314

315 4 Display Architectures

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

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

318 Display Device. Used to show image data.

319 Display Driver. May be one or more devices used to drive the display device.

320 Full-frame memory. Used to hold image data. Can be integrated in the display driver.

321 Registers. Used to configure display behavior and identification information. Can be integrated in
322 the display driver.

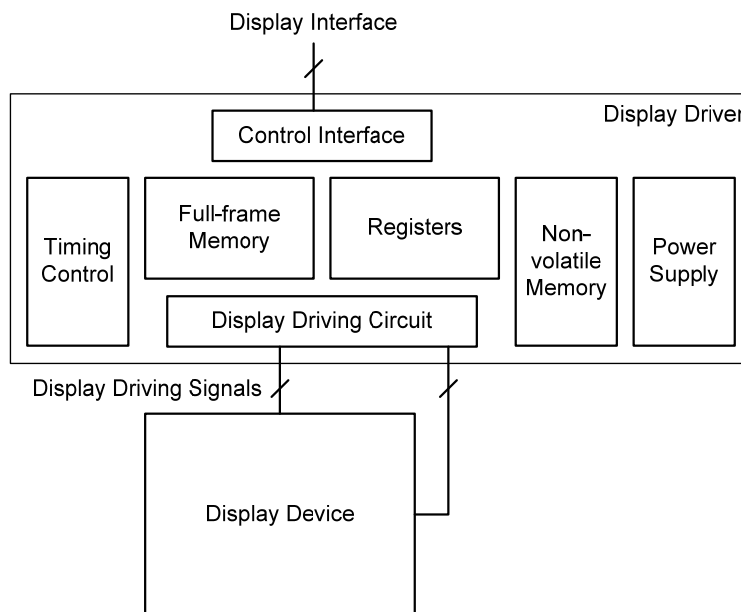
323 Timing Controller. Provides timing signals to control the display and display driver based on
324 configuration information. Can be integrated in the display driver.

325 Non-volatile memory. Used to store default register and configuration values. Can be integrated in
326 the display driver.

327 Control Interface. Provides the interface between the host processor and the display driver. Can be
328 integrated in the display driver.

329 Display Driving Circuit. Used to convert timing signals and voltages to signals appropriate to
330 drive the display device.

331 Power Supply. Used to convert system voltages to levels usable by the display device and display
332 driver. Can be integrated in the display driver.

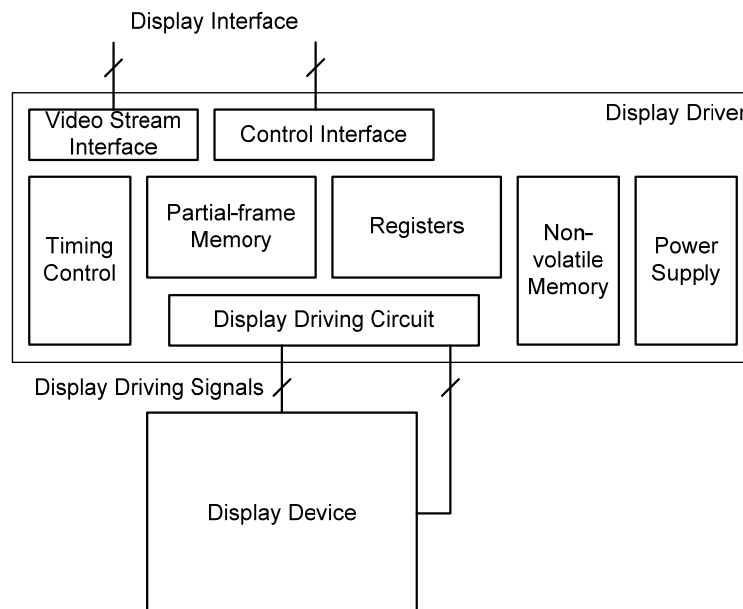


333

334

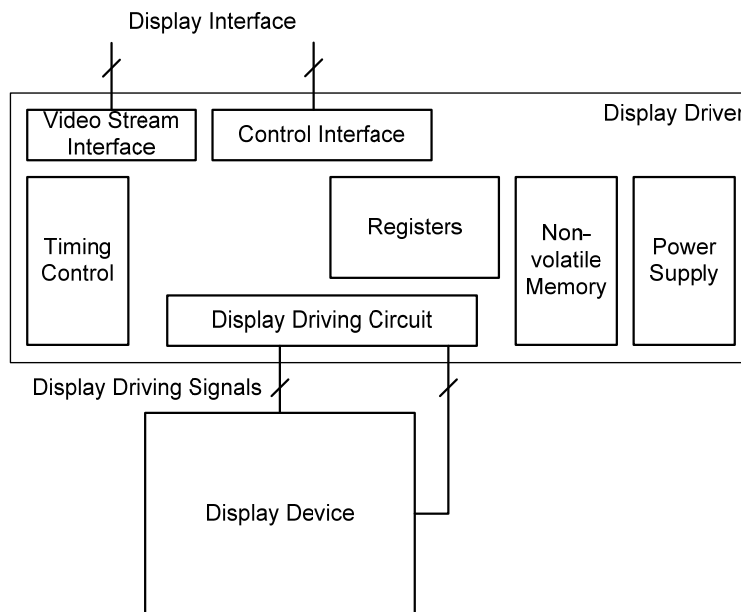
Figure 1 Type 1 Display Architecture Block Diagram

- 335 The Type 2 Display Architecture should consist of the following functional blocks:
- 336 Display Device. Used to show image data.
- 337 Display Driver. May be one or more devices used to drive the display device.
- 338 Partial-frame memory. Used to hold image data. Can be integrated in the display driver.
- 339 Registers. Used to configure display behavior and identification information. Can be integrated in
340 the display driver.
- 341 Timing Controller. Provides timing signals to control the display and display driver based on
342 configuration information. Can be integrated in the display driver.
- 343 Non-volatile memory. Used to store default register and configuration values. Can be integrated in
344 the display driver.
- 345 Control Interface. Provides the interface between the host processor and the display driver. Can be
346 integrated in the display driver.
- 347 Display Driving Circuit. Used to convert timing signals and voltages to signals appropriate to
348 drive the display device.
- 349 Power Supply. Used to convert system voltages to levels usable by the display device and display
350 driver. Can be integrated in the display driver.
- 351 Video Stream Interface. Used to receive video image data and timing signals from the host
352 processor.



353
354 **Figure 2 Type 2 Display Architecture Block Diagram**
355

- 356 The Type 3 Display Architecture should consist of the following functional blocks:
- 357 Display Device. Used to show image data.
- 358 Display Driver. May be one or more devices used to drive the display device.
- 359 Registers. Used to configure display behavior and identification information. Can be integrated in
360 the display driver.
- 361 Timing Controller. Provides timing signals to control the display and display driver based on
362 configuration information. Can be integrated in the display driver.
- 363 Non-volatile memory. Used to store default register and configuration values. Can be integrated in
364 the display driver.
- 365 Control Interface. Provides the interface between the host processor and the display driver. Can be
366 integrated in the display driver.
- 367 Display Driving Circuit. Used to convert timing signals and voltages to signals appropriate to
368 drive the display device.
- 369 Power Supply. Used to convert system voltages to levels usable by the display device and display
370 driver. Can be integrated in the display driver.
- 371 Video Stream Interface. Used to receive video image data and timing signals from the host
372 processor.



373
374

Figure 3 Type 3 Display Architecture Block Diagram

- 375 In all architecture types, it is assumed the power supply is under the control of the display driver.
- 376 The Display Command Set is used through the mentioned control interface.
- 377

378 **5 Display Functional Description**

379 **5.1 Power Level Definition**

380 Display modules designed using the Type 1 display architecture shall implement the power sequence
381 shown in Figure 4.

382 Display modules designed using the Type 2 display architecture shall implement the power sequence
383 shown in Figure 5.

384 Display modules designed using the Type 3 display architecture shall implement the power sequence
385 shown in Figure 6.

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

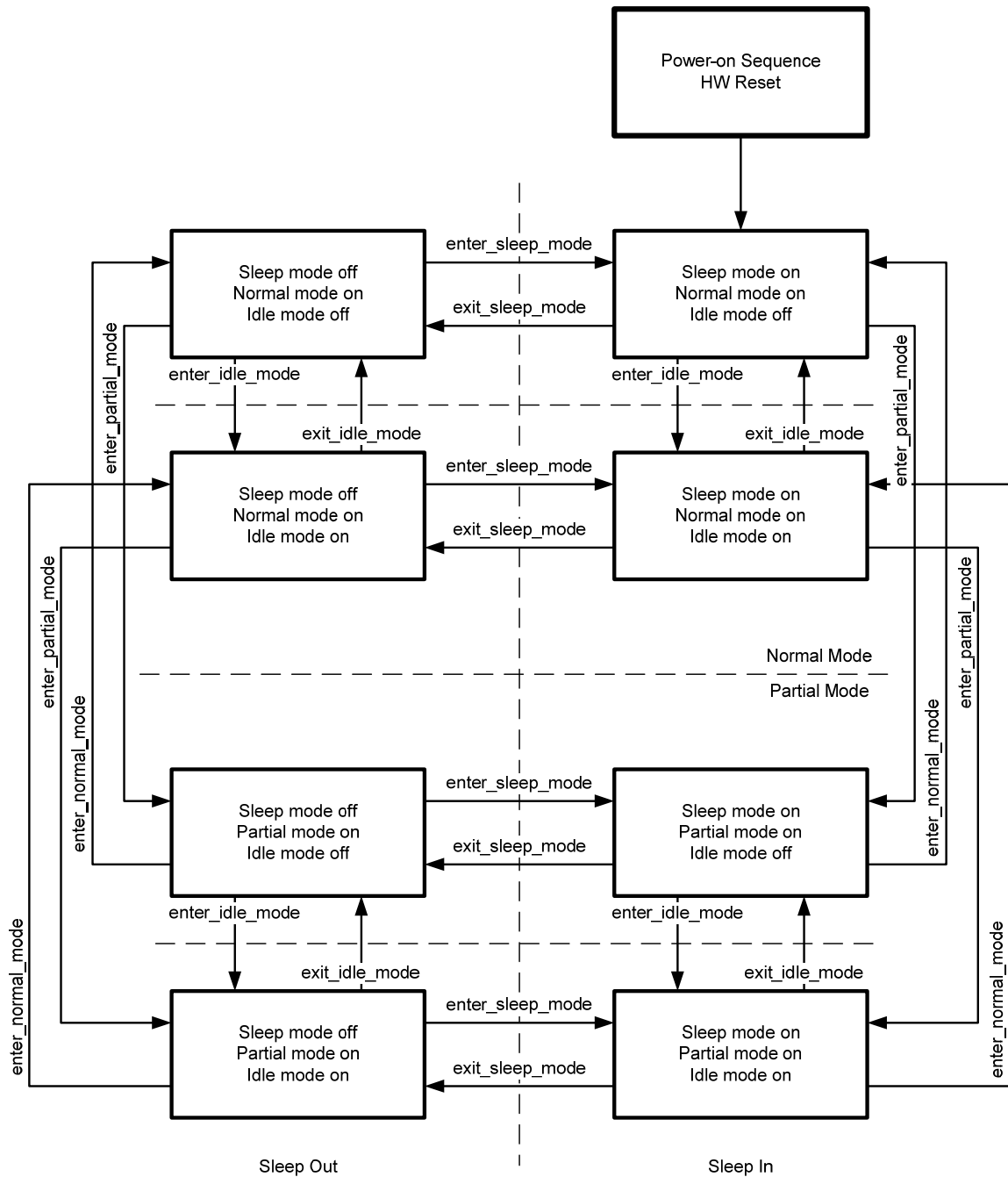
387 In Normal mode, the display module shows image data using the full display area of the display device. See
388 section 6.3 for a description of Normal mode.

389 In Partial mode, the display module shows image data in only a portion of the full display area of the
390 display device. See section 6.30 for a description of Partial mode.

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

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

398 When Sleep mode is off, the display module shows image data on the display device and all functional
399 blocks operate normally. See section 6.8 for a description of operation when Sleep mode is off.
400



401
402

403

Figure 4 Type 1 Display Architecture Power Change Sequences

404

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

405

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

406

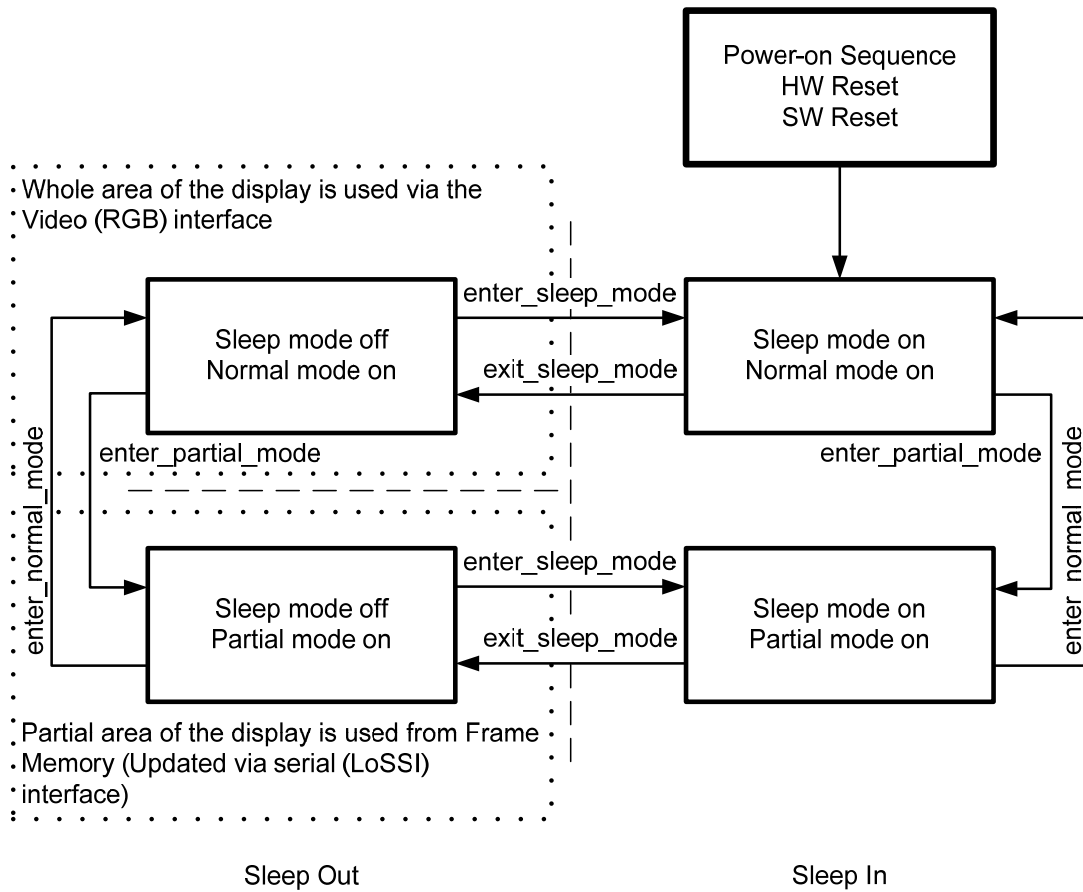


Figure 5 Type 2 Display Architecture Power Change Sequence

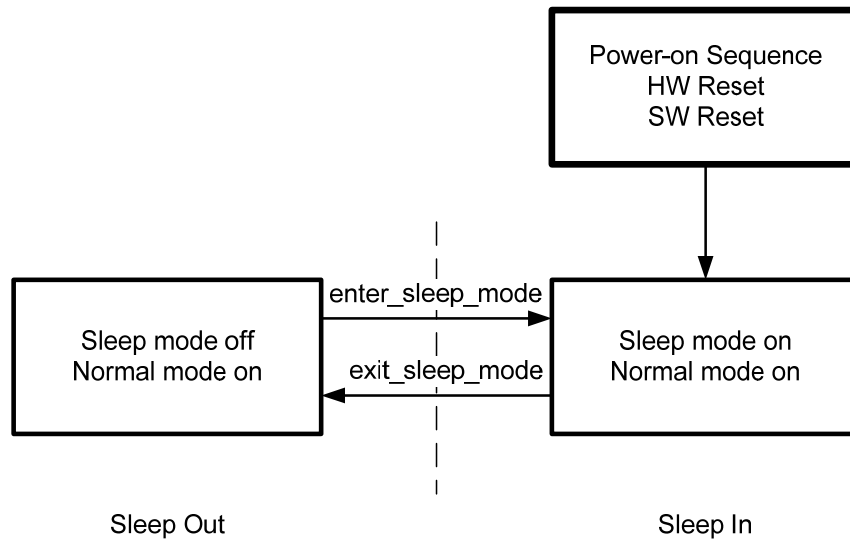


Figure 6 Type 3 Display Architecture Power Change Sequence

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

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

413

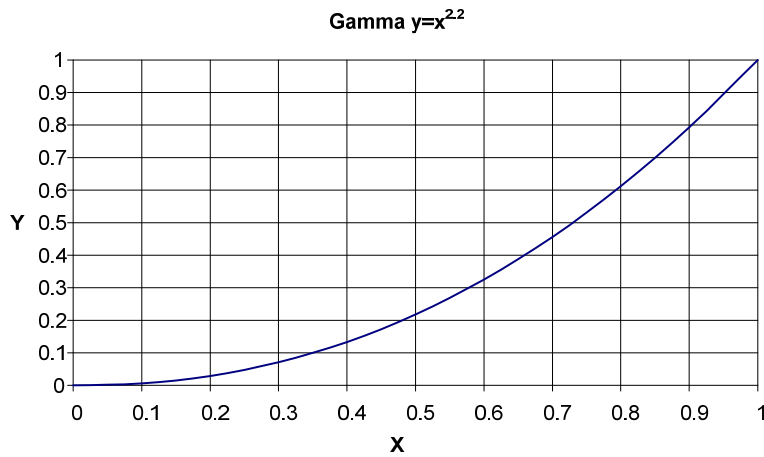
414 **5.2 Gamma Curves**

415 The display module can implement a gamma adjustment. If gamma adjustment is implemented then the
416 display module shall support at a minimum Gamma Curve 1 as described in section 5.2.1. The display
417 module can also implement up to three additional gamma curves as described in sections 5.2.2
418 through 5.2.4.

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

421 **5.2.1 Gamma Curve 1 (GC0)**

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



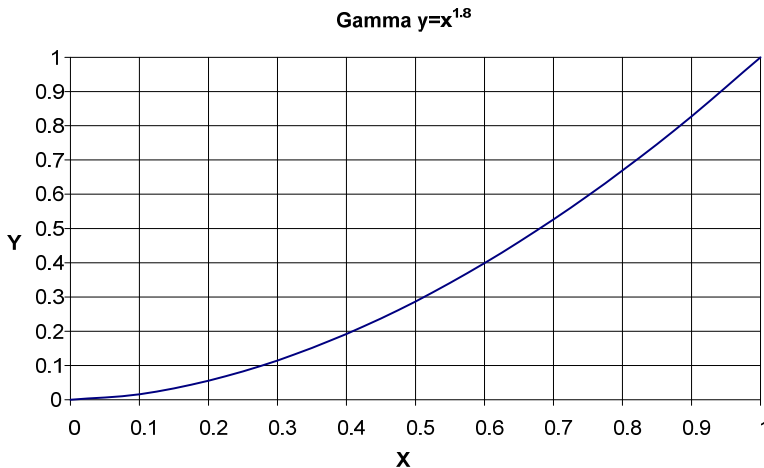
423

424

Figure 7 Gamma curve 1 (GC0)

425 **5.2.2 Gamma Curve 2 (GC1)**

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



427

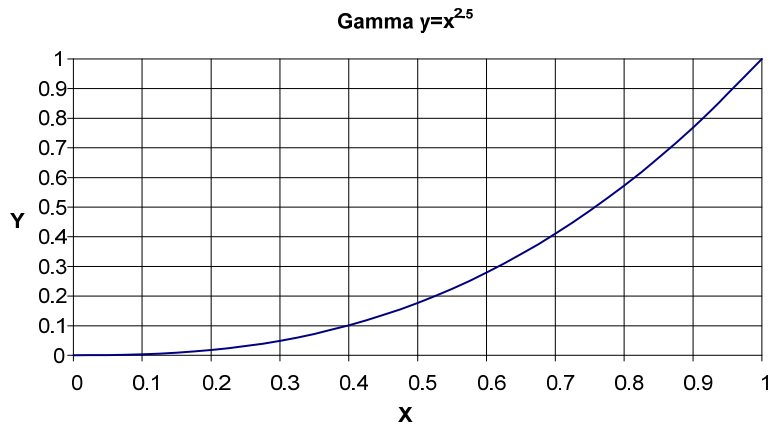
428

429

Figure 8 Gamma Curve 2 (GC1)

430 **5.2.3 Gamma Curve 3 (GC2)**

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



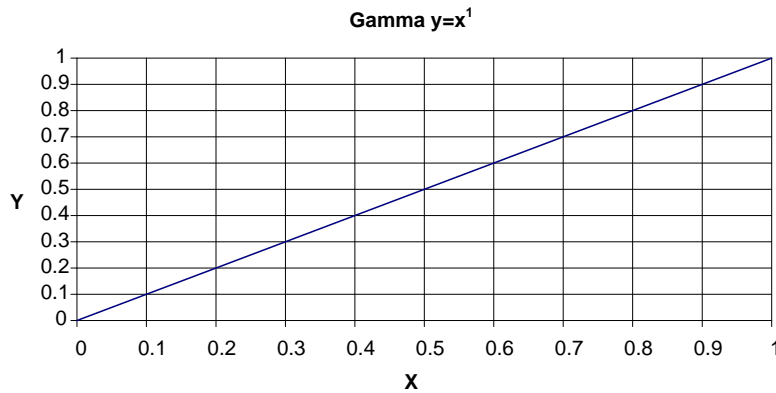
432

433

Figure 9 Gamma Curve 3 (GC2)

434 **5.2.4 Gamma Curve 4 (GC3)**

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



436

437

Figure 10 Gamma Curve 4 (GC3)

438

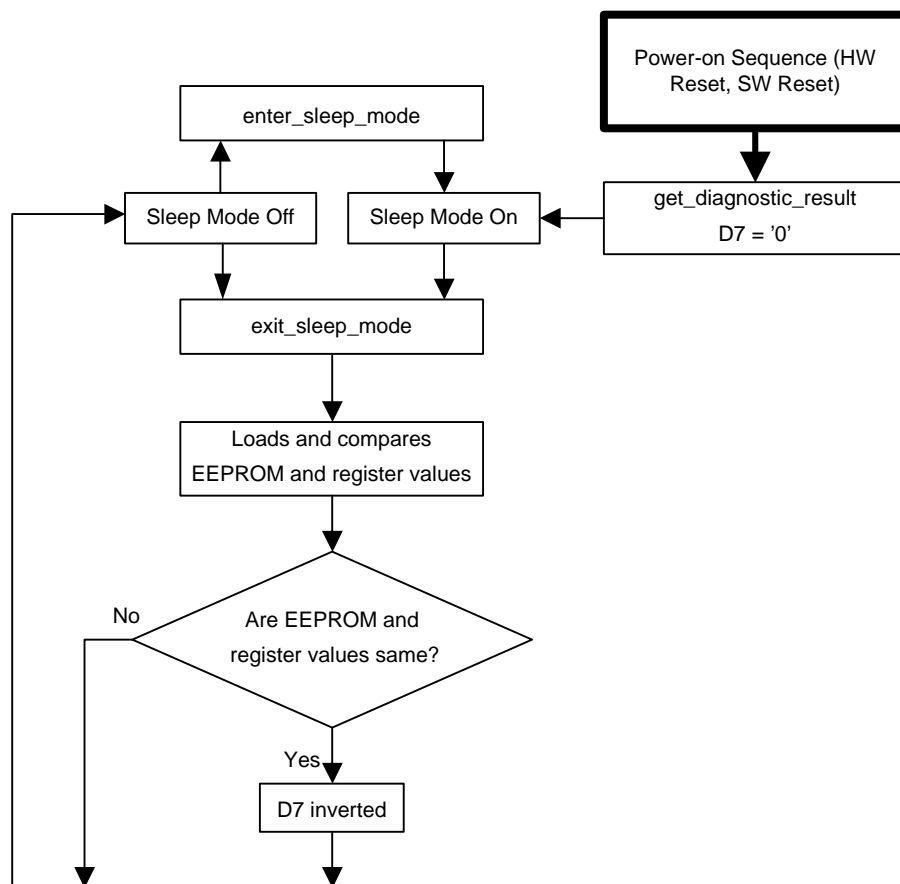
439 **5.3 Self-diagnostic Functions**

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

443 **5.3.1 Register Loading Detection**

444 The `exit_sleep_mode` command (see section 6.8) is a trigger for the Register Loading Detection function.
 445 This function indicates if the display module correctly loaded the factory default values from Non-volatile
 446 memory to the registers. If the registers were loaded properly then bit D7 of the SDR register is inverted,
 447 otherwise the value is unchanged. See section 6.11 for a description of the SDR register.

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



449

450

Figure 11 Register Loading Detection Flow Chart

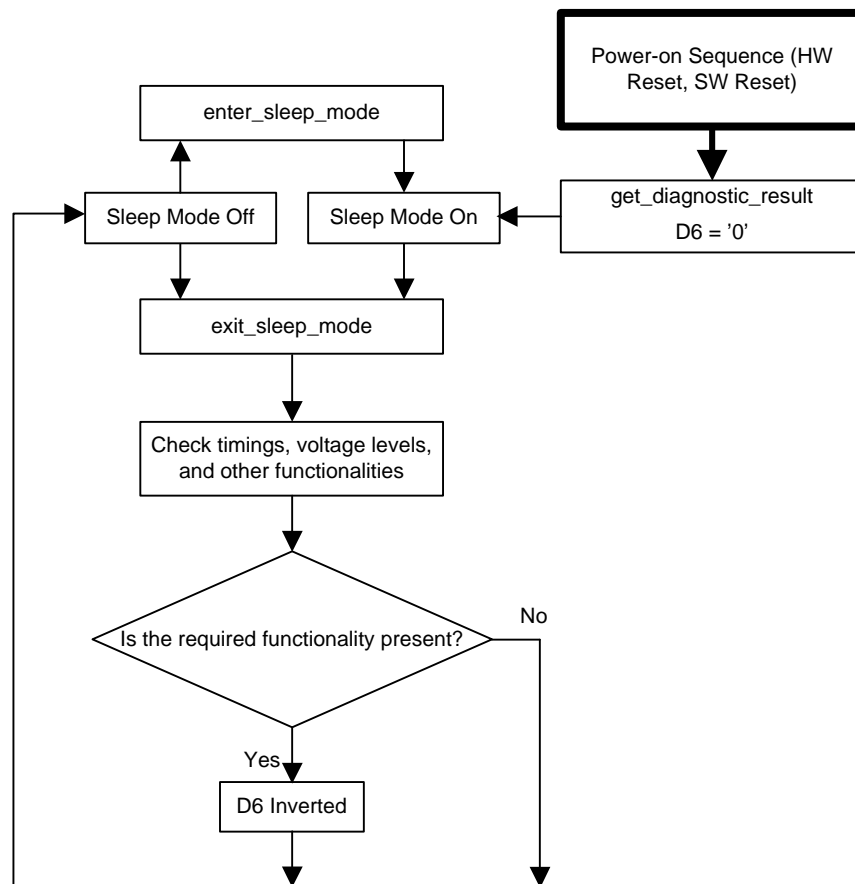
451 Note: Registers modified by the display module after loading are not verified.

452

453 **5.3.2 Functionality Detection**

454 The `exit_sleep_mode` command (see section 6.8) is a trigger for the Functionality Detection function. This
 455 function indicates if the display module functional blocks, e.g. power supply, clock generator, etc. are
 456 operating correctly. If the functional blocks are operating properly then bit D6 of the SDR register is
 457 inverted, otherwise the value is unchanged. See section 6.11 for a description of the SDR register.

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



459

460

Figure 12 Functionality Detection Flow Chart

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

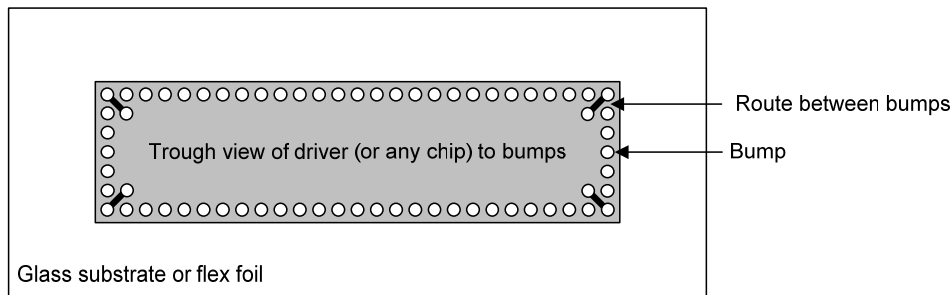
463

464 **5.3.3 Chip Attachment Detection (optional)**

465 The `exit_sleep_mode` command (see section 6.8) is a trigger for the Chip Attachment Detection function.
 466 This function indicates if certain chips, e.g. display driver IC, are attached to the display module. If the
 467 chips are properly attached to the display module then bit D5 of the SDR register is inverted, otherwise the
 468 value is unchanged. See section 6.11 for a description of the SDR register.

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

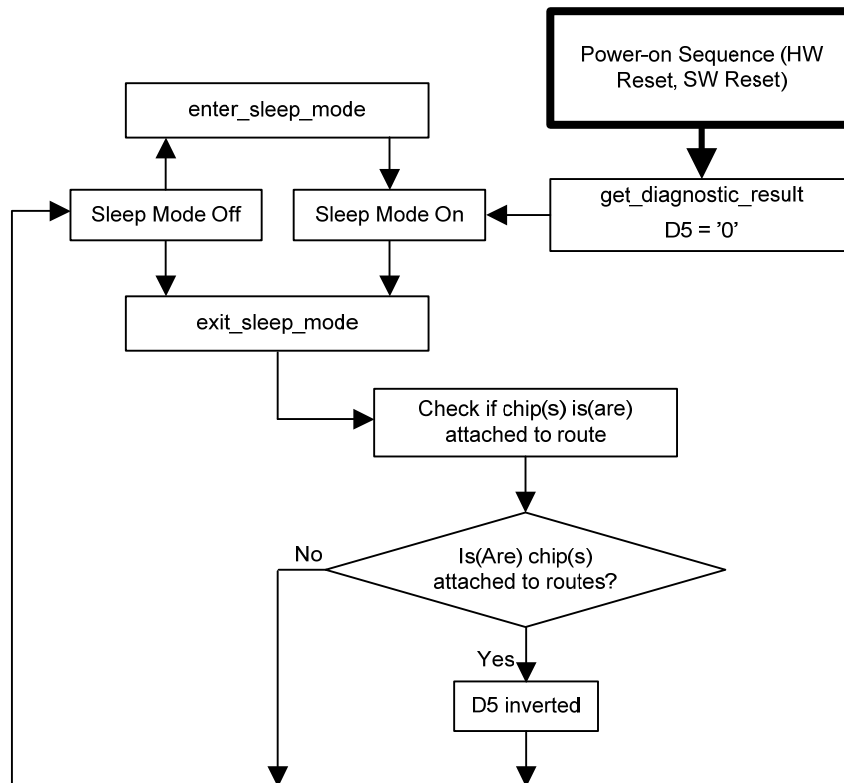
470 Figure 13 is a reference implementation for the Chip Attachment Detection function. Two bumps are
 471 connected together via a conductor on the flex foil or the display glass substrate in all four corners of the
 472 chip.



473

474

Figure 13 Chip Attachment Detection Reference



475

476

477

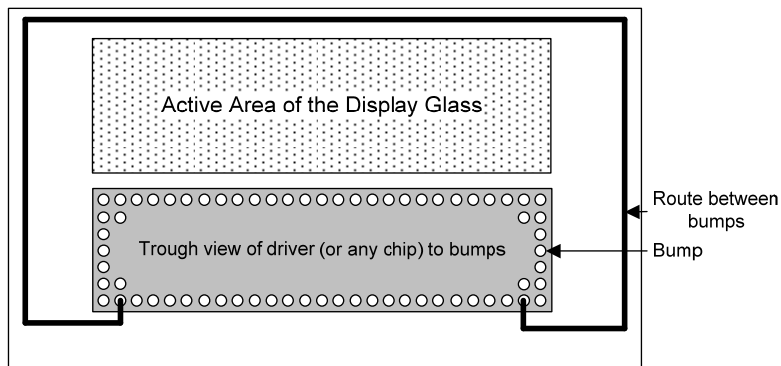
Figure 14 Chip Attachment Detection Flow Chart

478 **5.3.4 Display Glass Break Detection (optional)**

479 The `exit_sleep_mode` command (see section 6.8) is a trigger for the Display Glass Break Detection
 480 function. This function indicates if display glass is broken. If the display glass is broken then bit D4 of the
 481 SDR register is inverted, otherwise the value is unchanged. See section 6.11 for a description of the SDR
 482 register.

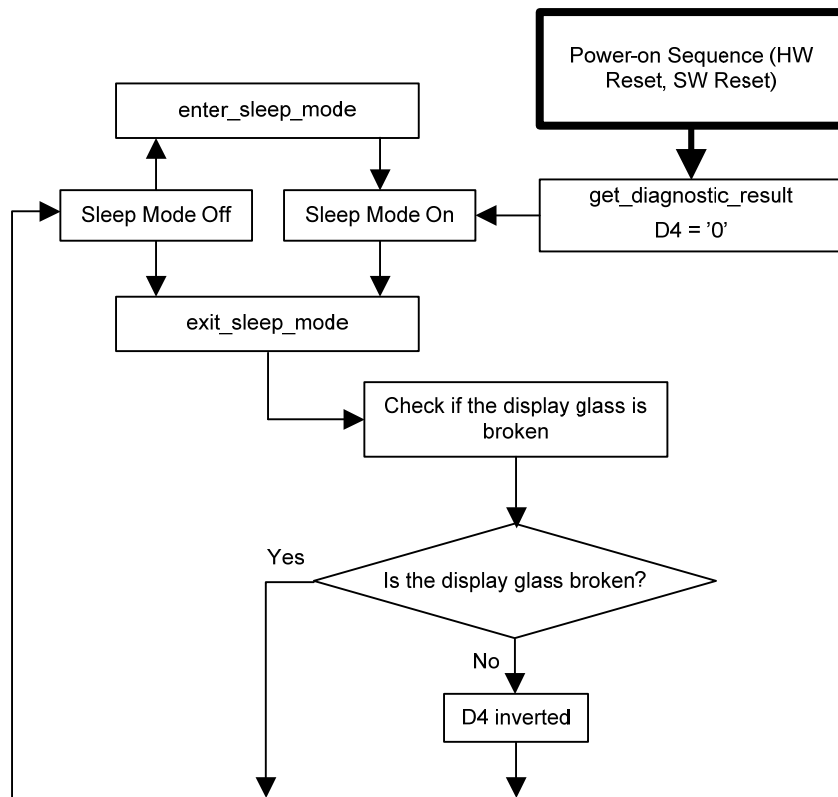
483 The flow chart for the Display Glass Break Detection function is shown in Figure 16.

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



486
487

Figure 15 Display Glass Break Detection Reference



488
489

Figure 16 Display Glass Break Detection Flow Chart

490 **5.4 Display Command Set**

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

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

497 The Manufacturer Command Set (MCS) is a device dependent interface intended for factory programming
498 of the display module default parameters. Once the display module is configured, the MCS shall be
499 disabled by the manufacturer. Once disabled, all MCS commands are treated as nop by the display
500 interface. The MCS is not defined in this specification.

501 The User Command Set provides a display device independent interface targeted at the operating system's
502 hardware abstraction layer. All commands listed in this section shall be implemented except write_LUT
503 which is optional.

504 Any unused command codes shall be treated as nop by the display module.

505 The remainder of this section is divided into three sections. Section 5.5 is an alphabetical list of the
506 supported commands. Section 5.6 and 5.7 describe command functionality in different display architectures
507 and operating modes.
508

509 **5.5 Command List**

510

Table 1 Command List

| 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_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_diagnostic_result | 0Fh | Get Peripheral Self-Diagnostic Result | 1 | Yes | Yes | Yes |
| get_display_mode | 0Dh | Get the current display mode from the peripheral. | 1 | Yes | Yes | Yes |
| get_green_channel | 07h | Get the green component of the pixel at (0, 0). | 1 | No | 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_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 |

| Command | Hex Code | Description | Number of Parameters | Display Architecture Implementation Requirement | | |
|----------------------|----------|---|----------------------|---|--------|--------|
| | | | | Type 1 | Type 2 | Type 3 |
| 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 |
| 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 |
| set_column_address | 2Ah | Set the column extent. | 4 | Yes | Yes | No |
| 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 |

| Command | Hex Code | Description | Number of Parameters | Display Architecture Implementation Requirement | | |
|-----------------------|----------|--|----------------------|---|--------|--------|
| | | | | Type 1 | Type 2 | Type 3 |
| 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 |
| soft_reset | 01h | Software Reset | 0 | Yes | Yes | Yes |
| 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 |

512 **5.6 Command Accessibility**

513 Table 2 provides command accessibility of several combinations of display and power modes.

514

Table 2 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_address_mode | 0Bh | Yes | Yes | Yes | Yes | Yes |
| get_blue_channel | 08h | Yes | Yes | N/A | N/A | Yes |
| get_diagnostic_result | 0Fh | Yes | Yes | Yes | Yes | Yes |
| get_display_mode | 0Dh | Yes | Yes | Yes | Yes | Yes |
| get_green_channel | 07h | Yes | Yes | N/A | N/A | Yes |
| get_pixel_format | 0Ch | Yes | Yes | Yes | Yes | Yes |
| get_power_mode | 0Ah | 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 |
| read_memory_continue | 3Eh | Yes | Yes | Yes | Yes | Yes |
| read_memory_start | 2Eh | Yes | Yes | Yes | Yes | Yes |
| set_address_mode | 36h | Yes | Yes | Yes | Yes | Yes |
| set_column_address | 2Ah | Yes | Yes | Yes | Yes | Yes |
| set_display_off | 28h | Yes | Yes | Yes | Yes | Yes |
| set_display_on | 29h | 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 |
| 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 |
| soft_reset | 01h | 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 |

515

516 **5.7 Default Modes and Values**

517 Table 3 provides default display modes, power modes and register values.

518 **Table 3 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_address_mode | 0Bh | 1 st | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) |
| get_blue_channel | 08h | 1 st | 00h | 00h | 00h |
| get_diagnostic_result | 0Fh | 1 st | 00h | 00h | 00h |
| get_display_mode | 0Dh | 1 st | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) |
| get_green_channel | 07h | 1 st | 00h | 00h | 00h |
| get_pixel_format | 0Ch | 1 st | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) |
| get_power_mode | 0Ah | 1 st | 08h | 08h | 08h |
| get_red_channel | 06h | 1 st | 00h | 00h | 00h |

| Command | Hex Code | Parameters | Default Modes and Values, Hex | | |
|----------------------|----------|-----------------------------------|--|--|--|
| | | | Power-on Sequence | SW Reset | HW Reset |
| get_scanline | 45h | 1 st & 2 nd | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) |
| get_signal_mode | 0Eh | 1 st | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) | Refer to corresponding command parameter(s) |
| 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 |
| set_address_mode | 36h | 1 st | 0000000b | No change from the value before SW reset | 0000000b |
| set_column_address | 2Ah | 1 st | 00h | 00h | 00h |
| | | 2 nd | 00h | 00h | 00h |
| | | 3 rd | 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 line. If set_address_mode's B5 = 1; The frame memory column address corresponding to the last horizontal line. | The frame memory column address corresponding to the last vertical line. |
| | | 4 th | | | |
| 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 | 1 st | 01h | 01h | 01h |
| set_page_address | 2Bh | 1 st | 00h | 00h | 00h |
| | | 2 nd | | | |

| Command | Hex Code | Parameters | Default Modes and Values, Hex | | |
|---------------------|----------|-----------------|--|--|--|
| | | | Power-on Sequence | SW Reset | HW Reset |
| | | 3 rd | 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. |
| | | 4 th | | | |
| set_partial_columns | 31h | 1 st | 00h | 00h | 00h |
| | | 2 nd | | | |
| | | 3 rd | 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 | 1 st | 00h | 00h | 00h |
| | | 2 nd | | | |
| | | 3 rd | 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. |
| | | 4 th | | | |
| set_pixel_format | 3Ah | 1 st | 07h | 07h | 07h |
| set_scroll_area | 33h | 1 st | 00h | 00h | 00h |
| | | 2 nd | 00h | 00h | 00h |
| | | 3 rd | 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. |
| | | 4 th | | | |
| | | 5 th | 00h | 00h | 00h |

| Command | Hex Code | Parameters | Default Modes and Values, Hex | | |
|-----------------------|----------|-----------------|-------------------------------|---------------------------|--------------------|
| | | | Power-on Sequence | SW Reset | HW Reset |
| | | 6 th | 00h | 00h | 00h |
| set_scroll_start | 37h | 1 st | 00h | 00h | 00h |
| | | 2 nd | 00h | 00h | 00h |
| set_tear_off | 34h | None | TE line output OFF | TE line output OFF | TE line output OFF |
| set_tear_on | 35h | 1 st | | | |
| set_tear_scanline | 44h | 1 st | 00h | 00h | 00h |
| | | 2 nd | 00h | 00h | 00h |
| soft_reset | 01h | None | N/A | N/A | N/A |
| 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 |

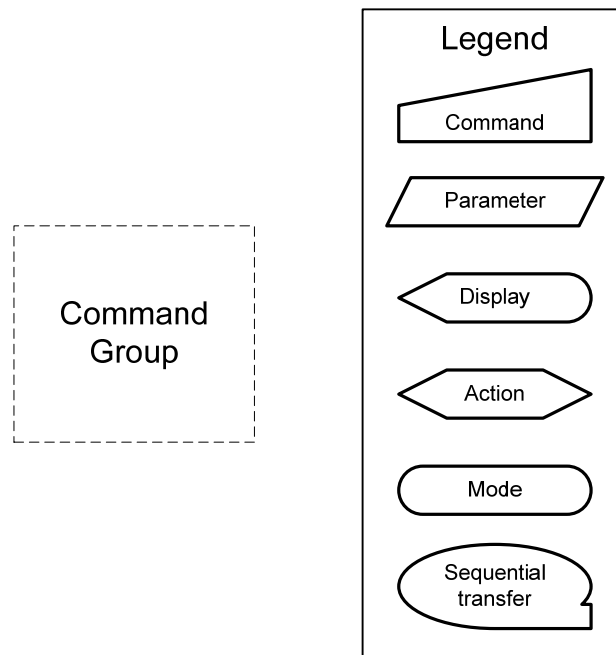
519

520 6 Command Description

521 This section defines the commands supported by display modules implementing MIPI Alliance
522 specifications for display interfaces.

523 All commands consist of a single 8-bit byte, in some cases accompanied by parameters that supply
524 necessary information for the correct execution of the command. Generally, the command and
525 accompanying parameter bytes are transferred using serial or parallel bits 0 through 7 of the display
526 interface, regardless of the physical interface width and architecture. The only exceptions are the
527 read_memory_continue, read_memory_start, write_memory_continue, and write_memory_start commands
528 in a DBI system (see [MIPI02]). The full width of the display interface may be used by these commands.
529 See sections 6.22, 6.23, 6.40, and 6.41 for the command descriptions.

530 Command flow charts in this section use the symbols defined in Figure 17.



531

532

Figure 17 Flowchart Legend

533

534 **6.1 enter_idle_mode**

535 **Interface** All
 536 **Command** 39h
 537 **Parameters** None

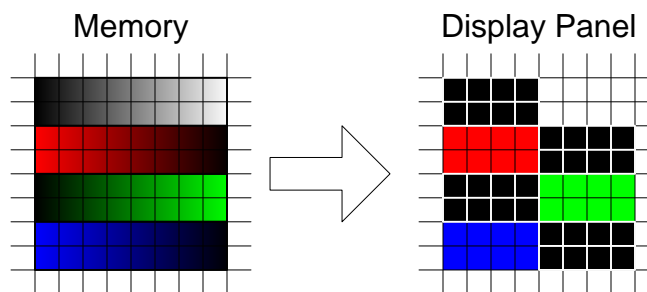
538 **Command**

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

539 **Description**

540 This command causes the display module to enter Idle Mode.

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



543

544

Figure 18 enter_idle_mode Example

545

546

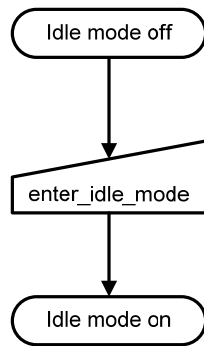
Table 4 enter_idle_mode Memory Content vs. Display Color

| Color | R₇ R₆ R₅ R₄ R₃ R₂ R₁ R₀ | G₇ G₆ G₅ G₄ G₃ G₂ G₁ G₀ | B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀ |
|--------------|--|--|--|
| Black | 0XXXXXXXX | 0XXXXXXXX | 0XXXXXXXX |
| Blue | 0XXXXXXXX | 0XXXXXXXX | 1XXXXXXXX |
| Red | 1XXXXXXXX | 0XXXXXXXX | 0XXXXXXXX |
| Magenta | 1XXXXXXXX | 0XXXXXXXX | 1XXXXXXXX |
| Green | 0XXXXXXXX | 1XXXXXXXX | 0XXXXXXXX |
| Cyan | 0XXXXXXXX | 1XXXXXXXX | 1XXXXXXXX |
| Yellow | 1XXXXXXXX | 1XXXXXXXX | 0XXXXXXXX |
| White | 1XXXXXXXX | 1XXXXXXXX | 1XXXXXXXX |

547 **Restrictions**

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

549 **Flow Chart**



550

551

552

Figure 19 enter_idle_mode Flow Chart

553 **6.2 enter_invert_mode**

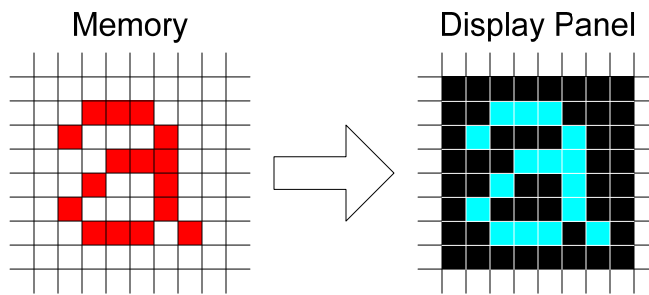
554 **Interface** All
 555 **Command** 21h
 556 **Parameters** None

557 **Command**

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

558 **Description**

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

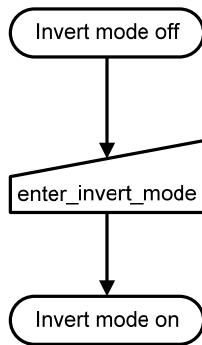


561
 562 **Figure 20 enter_invert_mode Example**

563 **Restrictions**

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

565 **Flow Chart**



566
 567 **Figure 21 enter_invert_mode Flow Chart**
 568

569 **6.3 enter_normal_mode**

570 **Interface** All
 571 **Command** 13h
 572 **Parameters** None

573 **Command**

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

574 **Description**

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

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

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

579 **Restrictions**

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

581 **Flow Chart**

582 See section 6.30 and section 6.33 for details of when to use this command.

583

584 **6.4 enter_partial_mode**

585 **Interface** All
 586 **Command** 12h
 587 **Parameters** None

588 **Command**

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

589 **Description**

590 This command causes the display module to enter the Partial Display Mode. The Partial Display Mode
 591 window is described by the set_partial_columns and set_partial_rows commands. See sections 6.30
 592 and 6.31, respectively, for details.

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

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

596 **Restrictions**

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

598 **Flow Chart**

599 See section 6.30.

600

601 **6.5 enter_sleep_mode**602 **Interface** All603 **Command** 10h604 **Parameters** None605 **Command**

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

606 **Description**

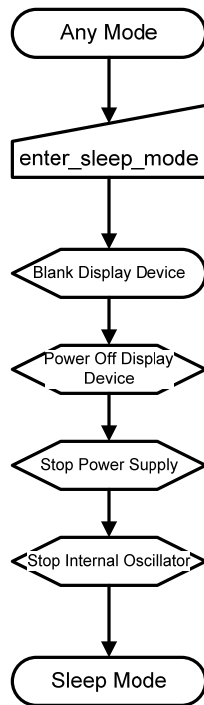
607 This command causes the display module to enter the Sleep mode.

608 In this mode, all unnecessary blocks inside the display module are disabled except interface
609 communication. This is the lowest power mode the display module supports.610 DBI or DSI Command Mode remains operational and the frame memory maintains its contents. The host
611 processor continues to send PCLK, HS and VS information to Type 2 and Type 3 display modules for two
612 frames after this command is sent when the display module is in Normal mode.613 **Restrictions**

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

615 The host processor must wait five milliseconds before sending any new commands to a display module
616 following this command to allow time for the supply voltages and clock circuits to stabilize.617 The host processor must wait 120 milliseconds after sending an exit_sleep_mode command before sending
618 an enter_sleep_mode command.

619 **Flow Chart**



620

621

622

Figure 22 enter_sleep_mode Flow Chart

623 **6.6 exit_idle_mode**624 **Interface** All625 **Command** 38h626 **Parameters** None627 **Command**

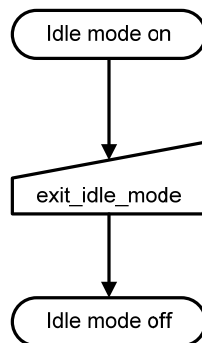
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hex Code |
|-------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------------|
| Direction H→D | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 38h |

628 **Description**

629 This command causes the display module to exit Idle mode.

630 **Restrictions**

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

632 **Flow Chart**

633

634

634 **Figure 23 exit_idle_mode Flow Chart**

635

636 **6.7 exit_invert_mode**

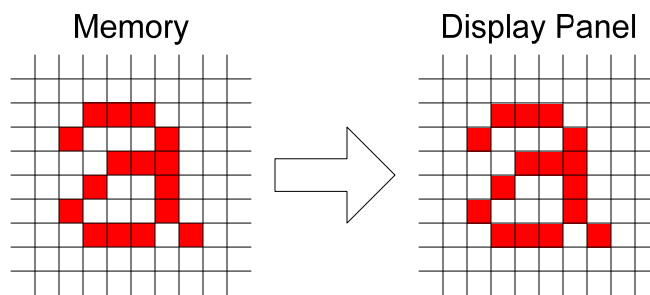
637 **Interface** All
 638 **Command** 20h
 639 **Parameters** None

640 **Command**

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

641 **Description**

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



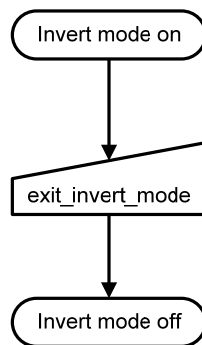
644
 645

Figure 24 exit_invert_mode Example

646 **Restrictions**

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

648 **Flow Chart**



649
 650
 651

Figure 25 exit_invert_mode Flow Chart

652 **6.8 exit_sleep_mode**

653 **Interface** All
 654 **Command** 11h
 655 **Parameters** None

656 **Command**

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

657 **Description**

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

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

662 **Restrictions**

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

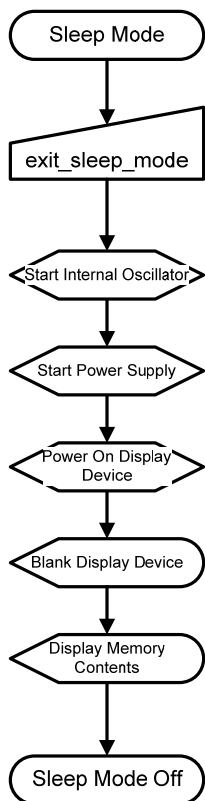
665 The host processor must wait five milliseconds after sending this command before sending another
 666 command. This delay allows the supply voltages and clock circuits to stabilize.

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

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

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

674 **Flow Chart**



675
676
677

Figure 26 exit_sleep_mode Flow Chart

678 **6.9 get_address_mode**679 **Interface** All680 **Command** 0Bh681 **Parameters** See below682 **Command**

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

683 **Parameter**

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

684 **Description**

685 The display module returns the current status.

686 Bit D7 – Page Address Order

687 ‘0’ = Top to Bottom

688 ‘1’ = Bottom to Top

689 Bit D6 – Column Address Order

690 ‘0’ = Left to Right

691 ‘1’ = Right to Left

692 Bit D5 - Page/Column Order

693 ‘0’ = Normal Mode

694 ‘1’ = Reverse Mode

695 Bit D4 – Line Address Order

696 ‘0’ = LCD Refresh Top to Bottom

697 ‘1’ = LCD Refresh Bottom to Top

698 Bit D3 – RGB/BGR Order

699 ‘0’ = RGB

700 ‘1’ = BGR

701 Bit D2 – Display Data Latch Data Order

702 ‘0’ = LCD Refresh Left to Right

703 ‘1’ = LCD Refresh Right to Left

704 Not applicable for display modules scanned line by line

705 Bit D1 – Flip Horizontal

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

708 ‘0’ = Normal

709 ‘1’ = Flipped

710 Bit D0 – Flip Vertical

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

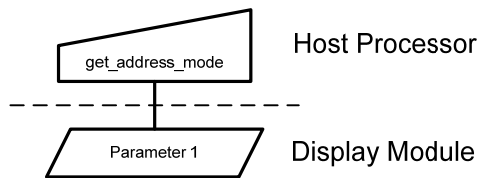
713 ‘0’ = Normal

714 ‘1’ = Flipped

715 **Restrictions**

716 None

717 **Flow Chart**



718

719

Figure 27 get_address_mode Flow Chart

720

721 **6.10 get_blue_channel**722 **Interface** All723 **Command** 08h724 **Parameters** See below725 **Command**

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

726

727 **Parameter**

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

728 **Description**

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

731 B7 is the MSB and B0 is the LSB.

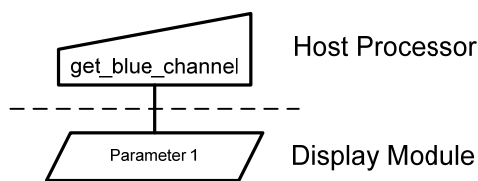
732 Only the relevant bits are used according to the pixel format; unused bits are set to '0'.

733 Examples:

- 734
- 12 bit format: B3 is MSB and B0 is LSB. B[7:4] are set to '0'.
 - 735 • 16 bit format: B5 is MSB, B1 is LSB and B7, B6 and B0 are set to '0'.
 - 736 • 18 bit format: B5 is MSB and B0 is LSB. B7 and B6 are set to '0'.
 - 737 • 24 bit format: B7 is MSB and B0 is LSB. All bits are used.

738 **Restrictions**

739 None

740 **Flow Chart**

741

742

743 **Figure 28 get_blue_channel Flow Chart**

743

744 **6.11 get_diagnostic_result**745 **Interface** All746 **Command** 0Fh747 **Parameters** See below748 **Command**

| | Direction | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hex Code |
|-----|------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------------|
| 749 | H→D | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0Fh |

750 **Parameter**

| | Direction | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hex Code |
|-----|------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------------|
| 751 | D→H | D7 | D6 | D5 | D4 | 0 | 0 | 0 | 0 | XXh |

751 **Description**

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

754 Bit D7 – Register Loading Detection

755 Bit D6 – Functionality Detection

756 Bit D5 – Chip Attachment Detection

757 Set to ‘0’ if feature unimplemented.

758 Bit D4 – Display Glass Break Detection

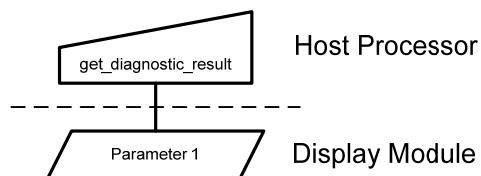
759 Set to ‘0’ if feature unimplemented.

760 Bits D[3:0] – Reserved

761 Set to ‘0’.

762 **Restrictions**

763 None

764 **Flow Chart**

765

766

767 **Figure 29 get_diagnostic_result Flow Chart**

767

768 **6.12 get_display_mode**769 **Interface** All770 **Command** 0Dh771 **Parameters** See below772 **Command**

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

773

774 **Parameter**

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

775 **Description**

776 The display module returns the Display Image Mode status.

777 Bit D7 – Vertical Scrolling Status

778 ‘0’ = Vertical Scrolling is Off.

779 ‘1’ = Vertical Scrolling is On.

780 Bit D6 – Reserved

781 Set to ‘0’.

782 Bit D5 – Inversion On/Off

783 ‘0’ = Inversion is Off.

784 ‘1’ = Inversion is On.

785 Bit D4 – Reserved

786 Set to ‘0’.

787 Bit D3 – Reserved

788 Set to ‘0’.

789 Bits D[2:0] – Gamma Curve Selection

790

Table 5 Gamma Curve Selection

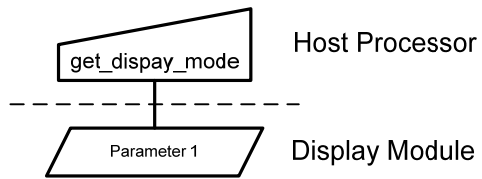
| Gamma Curve Selection | D2 | D1 | D0 | Gamma Set (26h) Parameter |
|------------------------------|-----------|-----------|-----------|----------------------------------|
| Gamma Curve 1 | 0 | 0 | 0 | GC0 |
| Gamma Curve 2 | 0 | 0 | 1 | GC1 |
| Gamma Curve 3 | 0 | 1 | 0 | GC2 |
| Gamma Curve 4 | 0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

| Gamma Curve Selection | D2 | D1 | D0 | Gamma Set (26h) Parameter |
|-----------------------|----|----|----|---------------------------|
| Reserved | 1 | 0 | 0 | Reserved |
| Reserved | 1 | 0 | 1 | Reserved |
| Reserved | 1 | 1 | 0 | Reserved |
| Reserved | 1 | 1 | 1 | Reserved |

791 **Restrictions**

792 None

793 **Flow Chart**



794

795

Figure 30 get_display_mode Flow Chart

796

797 **6.13 get_green_channel**

798 **Interface** All
 799 **Command** 07h
 800 **Parameters** See below

801 **Command**

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

802 **Parameter**

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

803 **Description**

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

806 G7 is the MSB and G0 is the LSB.

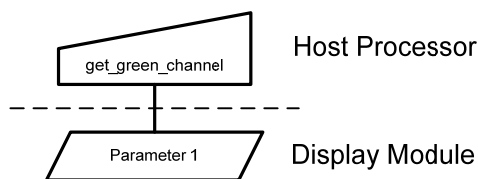
807 Only the relevant bits are used according to the pixel format; unused bits are set to '0'.

808 Examples:

- 809 • 12 bit format: G3 is MSB and G0 is LSB. G[7:4] are set to '0'.
- 810 • 16 bit format: G5 is MSB and G0 is LSB. G7 and G6 are set to '0'.
- 811 • 18 bit format: G5 is MSB and G0 is LSB. G7 and G6 are set to '0'.
- 812 • 24 bit format: G7 is MSB and G0 is LSB. All bits are used.

813 **Restrictions**

814 None

815 **Flow Chart**

816

817

Figure 31 get_green_channel Flow Chart

818

819 **6.14 get_pixel_format**

820 **Interface** All
 821 **Command** 0Ch
 822 **Parameters** See below

823 **Command**

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

824 **Parameter**

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

825 **Description**

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

827 Bits D[6:4] – DPI Pixel Format Definition

828 Bits D[2:0] – DBI Pixel Format Definition

829 Bits D7 and D3 are not used.

830 The pixel formats are shown in Table 6.

831 **Table 6 Interface Pixel Formats**

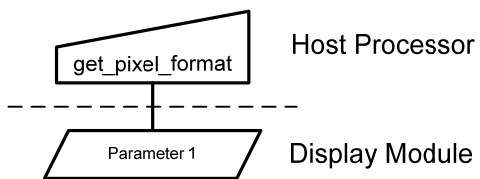
| Pixel Format | D6/D2 | D5/D1 | D4/D0 |
|---------------------|--------------|--------------|--------------|
| Reserved | 0 | 0 | 0 |
| 3 bits/pixel | 0 | 0 | 1 |
| 8 bits/pixel | 0 | 1 | 0 |
| 12 bits/pixel | 0 | 1 | 1 |
| Reserved | 1 | 0 | 0 |
| 16 bits/pixel | 1 | 0 | 1 |
| 18 bits/pixel | 1 | 1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

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

835 **Restrictions**

836 None

837 **Flow Chart**



838

839

Figure 32 get_pixel_format Flow Chart

840

841 **6.15 get_power_mode**842 **Interface** All843 **Command** 0Ah844 **Parameters** See below845 **Command**

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

846 **Parameter**

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

847 **Description**

848 The display module returns the current power mode.

849 Bit D7 – Reserved

850 Set to ‘0’

851 Bit D6 - Idle Mode On/Off

852 ‘0’ = Idle Mode Off.

853 ‘1’ = Idle Mode On.

854 Bit D5 – Partial Mode On/Off

855 ‘0’ = Partial Mode Off.

856 ‘1’ = Partial Mode On.

857 Bit D4 – Sleep Modet

858 ‘0’ = Sleep Mode On.

859 ‘1’ = Sleep Mode Off.

860 Bit D3 – Display Normal Mode On/Off

861 ‘0’ = Display Normal Mode Off.

862 ‘1’ = Display Normal Mode On.

863 Bit D2 – Display On/Off

864 ‘0’ = Display is Off.

865 ‘1’ = Display is On.

866 Bit D1 – Reserved

867 Set to ‘0’

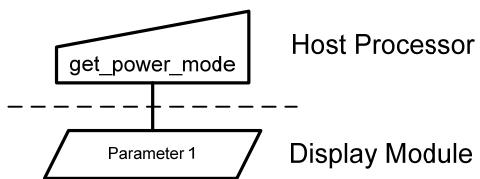
868 Bit D0 – Reserved

869 Set to ‘0’

870 **Restrictions**

871 None

872 **Flow Chart**



873

874

Figure 33 `get_power_mode` Flow Chart

875

876 **6.16 get_red_channel**
 877 **Interface** All
 878 **Command** 06h
 879 **Parameters** See below

880 **Command**

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

881 **Parameter**

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

882 **Description**

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

885 R7 is the MSB and R0 is the LSB.

886 Only the relevant bits are used according to the pixel format; unused bits are set to '0'

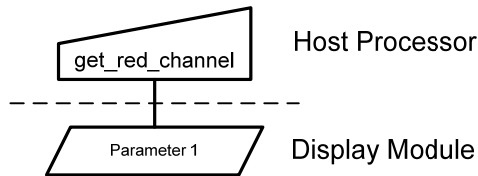
887 Examples:

- 888 • 12 bit format: R3 is MSB and R0 is LSB. R[7:4] are set to '0'.
- 889 • 16 bit format: R5 is MSB, R1 is LSB and R7, R6 and R0 are set to '0'.
- 890 • 18 bit format: R5 is MSB and R0 is LSB. R7 and R6 are set to '0'.
- 891 • 24 bit format: R7 is MSB and R0 is LSB. All bits are used.

892 **Restrictions**

893 None

894 **Flow Chart**



895
 896
 897

Figure 34 get_red_channel Flow Chart

898 **6.17 get_scanline**
 899 **Interface** All
 900 **Command** 45h
 901 **Parameters** See below

902 **Command**

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

903 **Parameter 1**

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

904 **Parameter 2**

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

905 **Description**

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

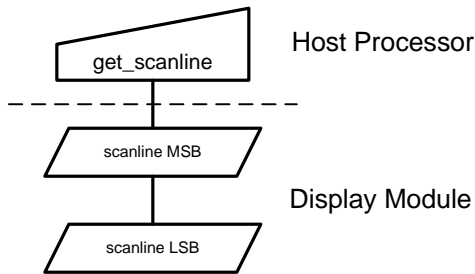
909 When in Sleep Mode, the value returned by get_scanline is undefined.

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

911 **Restrictions**

912 None

913 **Flow Chart**



914 **Figure 35 get_scanline Flow Chart**
 915
 916

917 **6.18 get_signal_mode**918 **Interface** All919 **Command** 0Eh920 **Parameters** See below921 **Command**

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

922 **Parameter**

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

923 **Description**

924 The display module returns the Display Signal Mode.

925 Bit D7 – Tearing Effect Line

926 ‘0’ = Tearing Effect Line Off.

927 ‘1’ = Tearing Effect On.

928 Bit D6 – Tearing Effect Line Output Mode.

929 See [MIPI02] and section 6.36 for mode definitions.

930 ‘0’ = Mode 0.

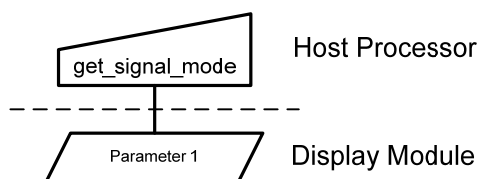
931 ‘1’ = Mode 1.

932 Bit D[5:0] – Reserved

933 Set to ‘0’.

934 **Restrictions**

935 None

936 **Flow Chart**

937

938

939 **Figure 36 get_signal_mode Flow Chart**

939

940 **6.19 nop**
 941 **Interface** All
 942 **Command** 00h
 943 **Parameters** None

944 **Command**

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

945 **Description**

946 This command does not have any effect on the display module. It can be used to terminate a Frame
 947 Memory Write or Read as described in the descriptions for write_memory_continue and
 948 read_memory_continue.

949 **Restrictions**

950 None

951 **Flow Chart**

952 None

953

954 **6.20 read_DDB_continue**

955 **Interface** All
 956 **Command** A8h
 957 **Parameters** See below

958 **Command**

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

959 **Parameter 1**

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

960
 961
 962

.

.

.

963 **Parameter N**

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

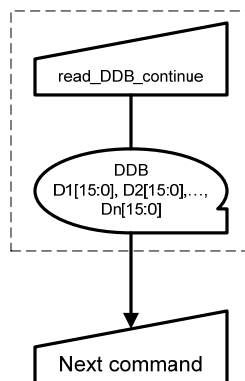
964 **Description**

965 See section 6.21.

966 **Restrictions**

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

969 **Flow Chart**



970
 971 **Figure 37 read_DDB_continue Flow Chart**
 972

973 **6.21 read_DDB_start**

974 **Interface** All
 975 **Command** A1h
 976 **Parameters** See below

977 **Command**

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

978 **Parameter 1**

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

979 **Parameter 2**

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

980 **Parameter 3**

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

981 **Parameter 4**

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

982 **Parameter 5**

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

983 **Description**

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

988 The format of returned data is as follows:

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

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

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

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

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

- 997
- FFh - Exit code – there is no more data in the Descriptor Block
- 998
- 00h - Escape code – there is supplier-proprietary data in the Descriptor Block (does not conform to any MIPI Alliance specification)
- 999
- Any other value – there is DDB data in the Descriptor Block.
- 1000

1001 DDBs may contain many more data fields providing information about the peripheral.

1002 In a DSI system, read activity takes the form of two separate transactions across the bus: first the read
1003 command `read_DDB_start` from host processor to peripheral, which includes the bus turn-around token.
1004 The peripheral then takes control of the bus and returns the requested data. The peripheral response to
1005 `read_DDB_start` is a Long Packet type, so its length may be up to 64K bytes unless limited by a previous
1006 `set_max_return_size` command.

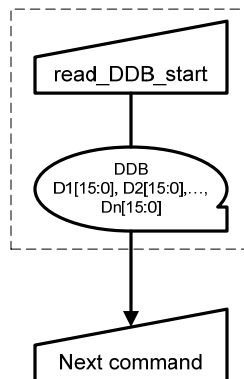
1007 The response to a `read_DDB_start` command always starts at the beginning of the Device Descriptor Block.
1008 After receiving the first packet and processing the returned DDB data, the host processor may initiate a
1009 `read_DDB_continue` command to access the next portion of the DDB. A `read_DDB_continue` command
1010 begins the next read at the location following the last byte of the previous data read from the DDB.

1011 Subsequent `read_DDB_continue` commands can be used to read a DDB or supplier-proprietary block of
1012 arbitrary size. There is, however, no obligation to read the entire block. The host processor may choose to
1013 stop reading after completion of any `read_DDB_xxx` command.

1014 **Restrictions**

1015 None

1016 **Flow Chart**



1017

1018

1019

Figure 38 read_DDB_start Flow Chart

1020 **6.22 read_memory_continue**

1021 **Interface** All
 1022 **Command** 3Eh
 1023 **Parameters** See below

1024 **Command**

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

1025 **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 |

1026

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

1027

1028

1029

.
.
.

1030 **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 |

1031

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

1032 **Description**

1033 This command transfers image data from the display module's frame memory to the host processor
 1034 continuing from the location following the previous read_memory_continue or read_memory_start
 1035 command.

1036 If set_address_mode B5 = 0:

1037 Pixels are read continuing from the pixel location after the read range of the previous read_memory_start or
 1038 read_memory_continue. The column register is then incremented and pixels are read from the frame
 1039 memory until the column register equals the End Column (EC) value. The column register is then reset to
 1040 SC and the page register is incremented. Pixels are read from the frame memory until the page register
 1041 equals the End Page (EP) value and the column register equals the EC value, or the host processor sends
 1042 another command.

1043 If set_address_mode B5 = 1:

1044 Pixels are read continuing from the pixel location after the read range of the previous read_memory_start or
 1045 read_memory_continue. The page register is then incremented and pixels are read from the frame memory
 1046 until the page register equals the End Page (EP) value. The page register is then reset to SP and the column
 1047 register is incremented. Pixels are read from the frame memory until the column register equals the End
 1048 Column (EC) value and the page register equals the EP value, or the host processor sends another
 1049 command.

1050 See section 6.25 for descriptions of the Start Column and End Column values.

1051 See section 6.29 for descriptions of the Start Page and End Page values.

1052 See section 8 in [MIPI01] and section 10 in [MIPI02] for color encoding for 8 or 9 bit image data.

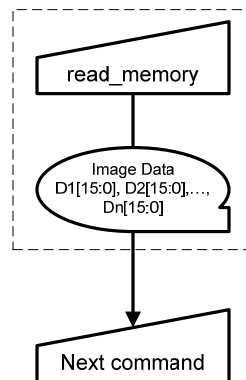
1053 Note the command description above shows 16-bit pixel data transferred over a 16-bit bus. Other
 1054 possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel
 1055 data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.

1056 **Restrictions**

1057 Regardless of the interface format chosen with the `set_pixel_format` command, the pixel format of the
 1058 returned data is always the maximum pixel depth supported by the display module. The display module
 1059 documentation shall describe the maximum pixel depth as well as the format of the data returned by the
 1060 display module when using this command.

1061 A `read_memory_start` should follow a `set_column_address`, `set_page_address` or `set_address_mode` to
 1062 define the read location. Otherwise, data read with `read_memory_continue` is undefined.

1063 **Flow Chart**



1064

1065

Figure 39 read_memory_continue Flow Chart

1066

1067 **6.23 read_memory_start**

1068 **Interface** All
 1069 **Command** 2Eh
 1070 **Parameters** See below

1071 **Command**

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

1072 **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 |

1073

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

1074

1075

1076

.
.
.

1077 **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 |

1078

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

1079 **Description**

1080 This command transfers image data from the display module's frame memory to the host processor starting
 1081 at the pixel location specified by preceding set_column_address and set_page_address commands.

1082 If set_address_mode B5 = 0:

1083 The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

1084 Pixels are read from frame memory at (SC, SP). The column register is then incremented and pixels read
 1085 from the frame memory until the column register equals the End Column (EC) value. The column register
 1086 is then reset to SC and the page register is incremented. Pixels are read from the frame memory until the
 1087 page register equals the End Page (EP) value and the column register equals the EC value, or the host
 1088 processor sends another command.

1089 If set_address_mode B5 = 1:

1090 The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

1091 Pixels are read from frame memory at (SC, SP). The page register is then incremented and pixels read from
 1092 the frame memory until the page register equals the End Page (EP) value. The page register is then reset to
 1093 SP and the column register is incremented. Pixels are read from the frame memory until the column register

1094 equals the End Column (EC) value and the page register equals the EP value, or the host processor sends
1095 another command.

1096 See section 6.25 for descriptions of the Start Column and End Column values.

1097 See section 6.29 for descriptions of the Start Page and End Page values.

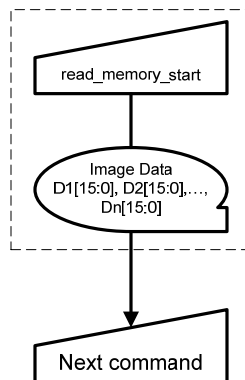
1098 See section 8 in [MIPI01] and section 10 in [MIPI02] for color encoding for 8 or 9 bit image data.

1099 Note the command description above shows 16-bit pixel data transferred over a 16-bit bus. Other
1100 possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel
1101 data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.

1102 **Restrictions**

1103 Regardless of the interface format chosen with the `set_pixel_format` command, the pixel format of the
1104 returned data is always the maximum pixel depth supported by the display module. The display module
1105 documentation shall describe the maximum pixel depth as well as the format of the data returned by the
1106 display module when using this command.

1107 **Flow Chart**



1108

1109

Figure 40 read_memory_start Flow Chart

1110

1111 **6.24 set_address_mode**

1112 **Interface** All
 1113 **Command** 36h
 1114 **Parameters** See below

1115 **Command**

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

1116 **Parameter**

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

1117 **Description**

1118 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].
 1119

1120 All bits are valid for peripherals based on the Type 2 display architecture operating in Command Mode, or
 1121 for peripherals based on the Type 1 display architecture. Bits B5, B4, B2, B1 and B0 have no effect on
 1122 peripherals based on the Type 2 display architecture operating in Video Mode, or for peripherals based on
 1123 the Type 3 display architecture.

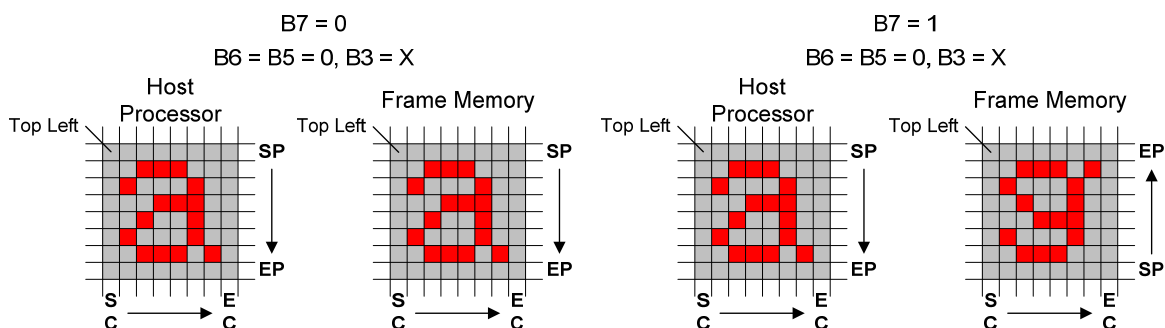
1124 No status bits are changed.

1125 Bit B7 – Page Address Order

1126 This bit controls the order that Pages of data are transferred from the host processor to the peripheral's
 1127 frame memory for a Type 1 or a Type 2 display architecture operating in Command Mode. This bit also
 1128 controls the Host processor to display device data latching order for a Type 2 or Type 3 display architecture
 1129 operating in Video Mode.

1130 '0' = Top to Bottom, Pages transferred from SP to EP

1131 '1' = Bottom to Top, Pages transferred from EP to SP



1132
 1133

1134

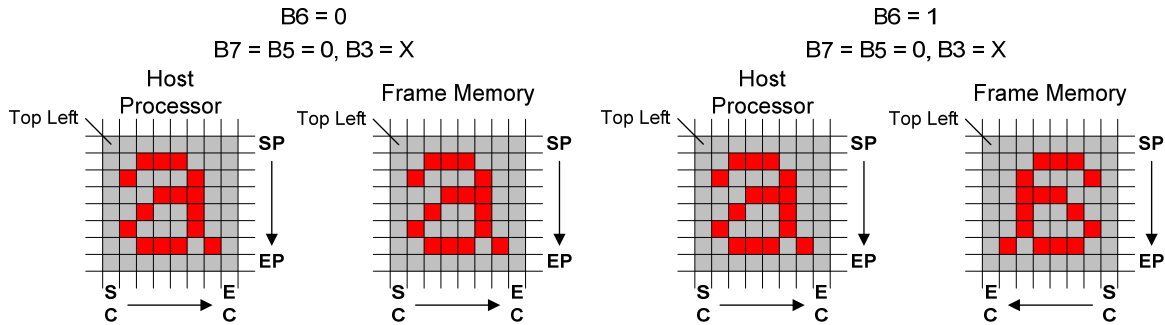
Figure 41 B7 Page Address Order

1135 Bit B6 – Column Address Order

1136 This bit controls the order that Columns of data are transferred from the host processor to the peripheral's
 1137 frame memory for a Type 1 or Type 2 display architecture operating in Command Mode. This bit also
 1138 controls the Host processor to display device data latching order for a Type 2 or Type 3 display architecture
 1139 operating in Video Mode.

1140 '0' = Left to Right, Columns transferred from SC to EC

1141 '1' = Right to Left, Columns transferred from EC to SC



1144 **Figure 42 B6 Column Address Order**

1145 Bit B5 – Page/Column Addressing Order

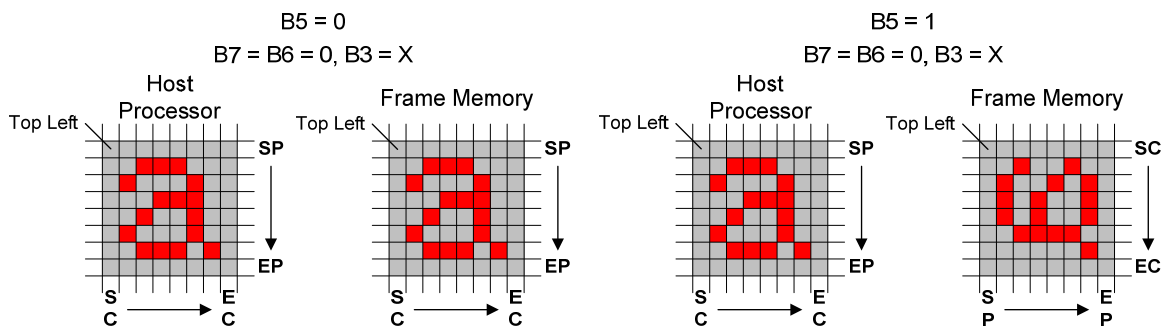
1146 This bit controls the order that Columns of data are transferred from the host processor to the peripheral's
 1147 frame memory.

1148 '0' = Normal Mode

1149 See section 6.41 (B5 = 0) for a description of Normal Mode operation.

1150 '1' = Reverse Mode

1151 See section 6.41 (B5 = 1) for a description of Reverse Mode operation.



1154 **Figure 43 B5 Page/Column Addressing Order**

1155 Bit B4 – Display Device Line Refresh Order

1156 This bit controls the display device’s horizontal line refresh order. The image shown on the display device
1157 is unaffected, regardless of the bit setting.

1158 ‘0’ = Display device is refreshed from the top line to the bottom line

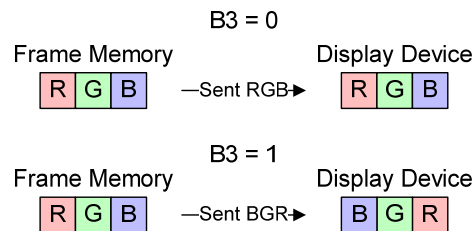
1159 ‘1’ = Display device is refreshed from the bottom line to the top line

1160 Bit B3 – RGB/BGR Order

1161 This bit controls the RGB data latching order transferred from the peripheral’s frame memory to the display
1162 device for a Type 1 or a Type 2 display architecture operating in Command Mode. This bit also controls the
1163 RGB data latching order transfer from the Host processor to the display device for a Type 2 or a Type 3
1164 display architecture operating in Video Mode.

1165 ‘0’ = Pixels sent in RGB order

1166 ‘1’ = Pixels sent in BGR order



1167

1168

Figure 44 B3 RGB Order

1169 Bit B2 – Display Data Latch Data Order

1170 This bit controls the display device’s vertical line data latch order. The image shown on the display device
1171 is unaffected, regardless of the bit setting.

1172 ‘0’ = Display device is refreshed from the left side to the right side

1173 ‘1’ = Display device is refreshed from the right side to the left side

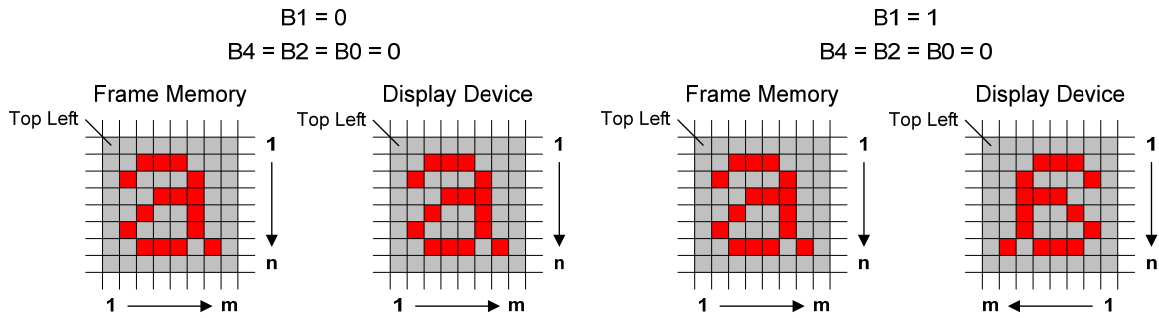
1174 Note: This bit has no visual effect if the display device is refreshed line by line.

1175 Bit B1 – Flip Horizontal

1176 This bit flips the image shown on the display device left to right. No change is made to the frame memory.

1177 ‘0’ = Normal

1178 ‘1’ = Flipped



1179
1180

Figure 45 B1 Flip Horizontal

1181

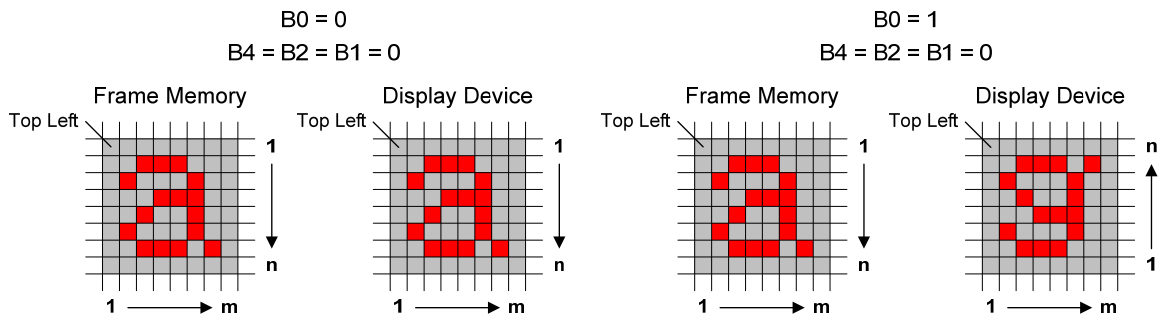
1182 Bit B0 – Flip Vertical

1183 This bit flips the image shown on the display device top to bottom by changing the gate scanning order.

1184 Neither the frame memory contents nor the order data is read from frame memory is changed.

1185 ‘0’ = Normal

1186 ‘1’ = Flipped



1187
1188

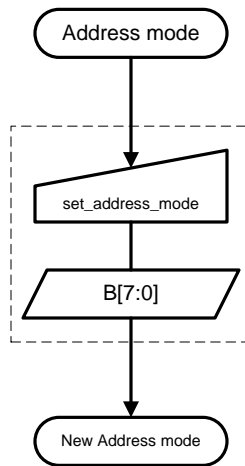
Figure 46 B0 Flip Vertical

1189

1190 **Restrictions**

1191 None

1192 **Flow Chart**



1193

1194

1195

Figure 47 set_address_mode Flow Chart

1196 **6.25 set_column_address**

1197 **Interface** All
 1198 **Command** 2Ah
 1199 **Parameters** See below

1200 **Command**

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

1201 **Parameter 1**

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

1202 **Parameter 2**

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

1203 **Parameter 3**

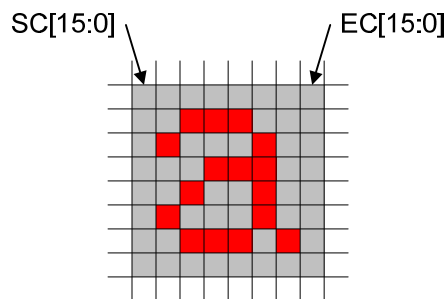
| Direction | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hex Code |
|-----------|------|------|------|------|------|------|-----|-----|----------|
| H→D | EC15 | EC14 | EC13 | EC12 | EC11 | EC10 | EC9 | EC8 | XXh |

1204 **Parameter 4**

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

1205 **Description**

1206 This command defines the column extent of the frame memory accessed by the host processor with the
 1207 read_memory_continue and write_memory_continue commands. No status bits are changed.



1208

1209

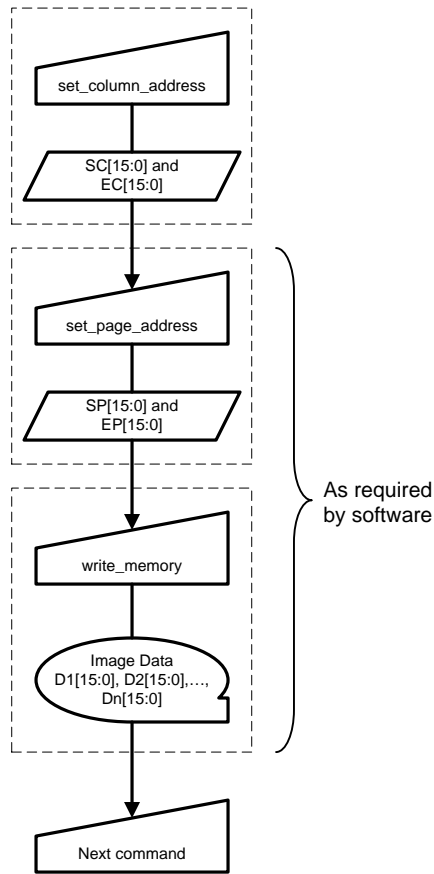
Figure 48 set_column_address Example

1210 **Restrictions**

1211 SC[15:0] must always be equal to or less than EC[15:0].

1212 If SC[15:0] or EC[15:0] is greater than the available frame memory then the parameter is not updated.

1213 **Flow Chart**



1214

1215

1216

Figure 49 set_column_address Flow Chart

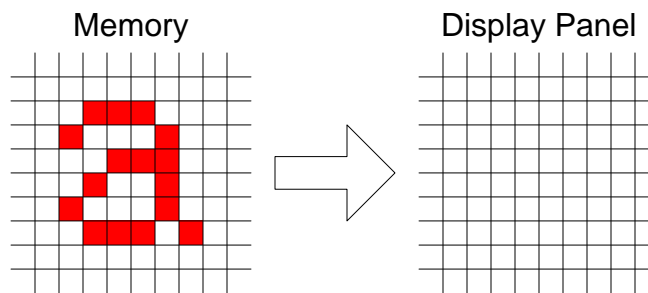
1217 **6.26 set_display_off**
 1218 **Interface** All
 1219 **Command** 28h
 1220 **Parameters** None

1221 **Command**

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

1222 **Description**

1223 This command causes the display module to stop displaying the image data on the display device. The
 1224 frame memory contents remain unchanged. No status bits are changed.



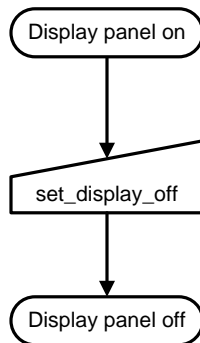
1225
 1226

Figure 50 set_display_off Example

1227 **Restrictions**

1228 This command has no effect when the display panel is already off.

1229 **Flow Chart**



1230
 1231
 1232

Figure 51 set_display_off Flow Chart

1233 **6.27 set_display_on**

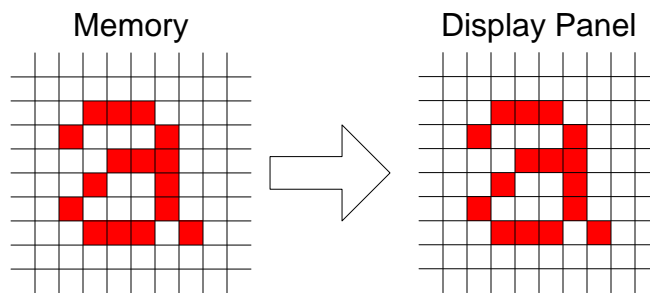
1234 **Interface** All
 1235 **Command** 29h
 1236 **Parameters** None

1237 **Command**

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

1238 **Description**

1239 This command causes the display module to start displaying the image data on the display device. The
 1240 frame memory contents remain unchanged. No status bits are changed.



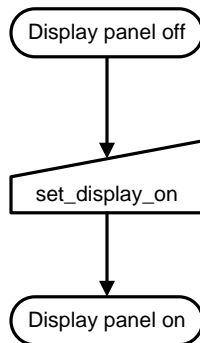
1241
 1242

Figure 52 set_display_on Example

1243 **Restrictions**

1244 This command has no effect when the display panel is already on.

1245 **Flow Chart**



1246
 1247
 1248

Figure 53 set_display_on Flow Chart

1249 **6.28 set_gamma_curve**

1250 **Interface** All
 1251 **Command** 26h
 1252 **Parameters** See below

1253 **Command**

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

1254 **Parameter**

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

1255 **Description**

1256 This command selects the desired gamma curve for the display device. Four fixed gamma curves are
 1257 defined in section 5.2. A curve is selected by setting the appropriate bit in the parameter as described in the
 1258 following table.

1259

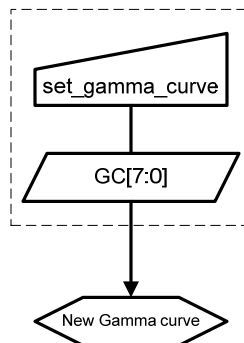
Table 7 Gamma Curves

| 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 |

1260 Note: All other values are reserved.

1261 **Restrictions**

1262 Values of GC[7:0] not shown in Table 7 above are reserved and shall not change the currently selected
 1263 gamma curve.

1264 **Flow Chart**

1265

1266

1267

Figure 54 set_gamma_curve Flow Chart

1268 **6.29 set_page_address**

1269 **Interface** All
 1270 **Command** 2Bh
 1271 **Parameters** See below

1272 **Command**

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

1273 **Parameter 1**

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

1274 **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 |

1275 **Parameter 3**

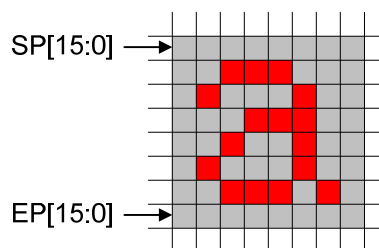
| Direction | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hex Code |
|-----------|------|------|------|------|------|------|-----|-----|----------|
| H→D | EP15 | EP14 | EP13 | EP12 | EP11 | EP10 | EP9 | EP8 | XXh |

1276 **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 |

1277 **Description**

1278 This command defines the page extent of the frame memory accessed by the host processor with the
 1279 write_memory_continue and read_memory_continue command. No status bits are changed.



1280

1281

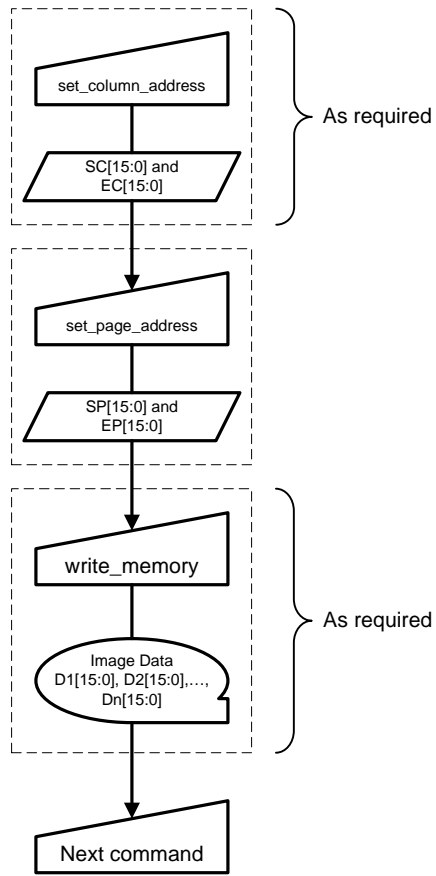
Figure 55 set_page_address Example

1282 **Restrictions**

1283 SP[15:0] must always be equal to or less than EP[15:0]

1284 If SP[15:0] or EP[15:0] is greater than the available frame memory then the parameter is not updated.

1285 **Flow Chart**



1286

1287

1288

Figure 56 set_page_address Flow Chart

1289 **6.30 set_partial_columns**

1290 **Interface** All
 1291 **Command** 31h
 1292 **Parameters** See below

1293 **Command**

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

1294 **Parameter 1**

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

1295 **Parameter 2**

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

1296 **Parameter 3**

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

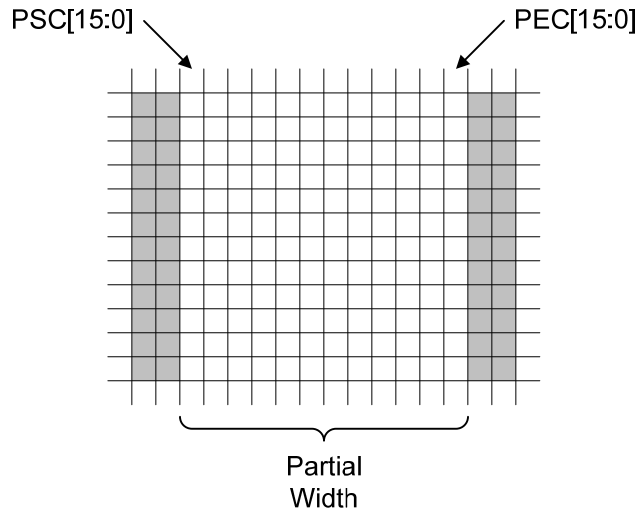
1297 **Parameter 4**

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

1298 **Description**

1299 This command defines the Partial Display mode's display width. There are two parameters associated with
 1300 this command, the first defines the Start Column (PSC) and the second the End Column (PEC), as
 1301 illustrated in Figure 57 through Figure 60. PSC and PEC refer to the Frame Memory Column Pointer.

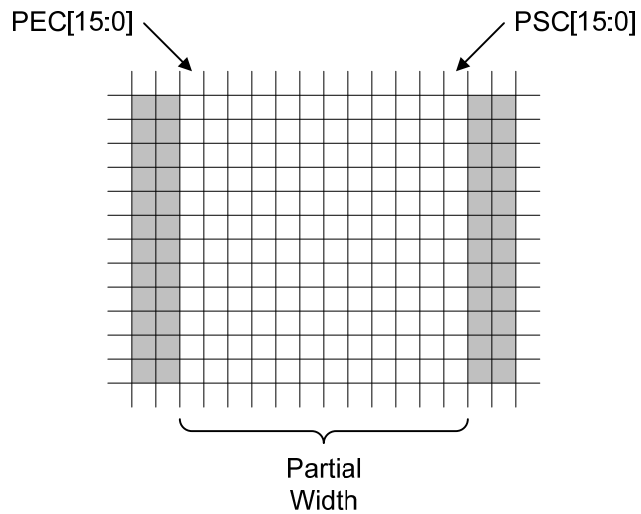
1302 If End Column > Start Column



1303

1304

Figure 57 set_partial_columns with set_address_mode B2 = 0

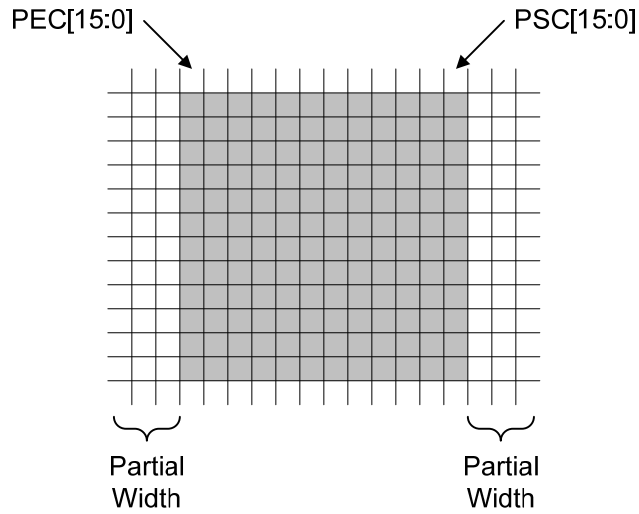


1305

1306

Figure 58 set_partial_columns with set_address_mode B2=1

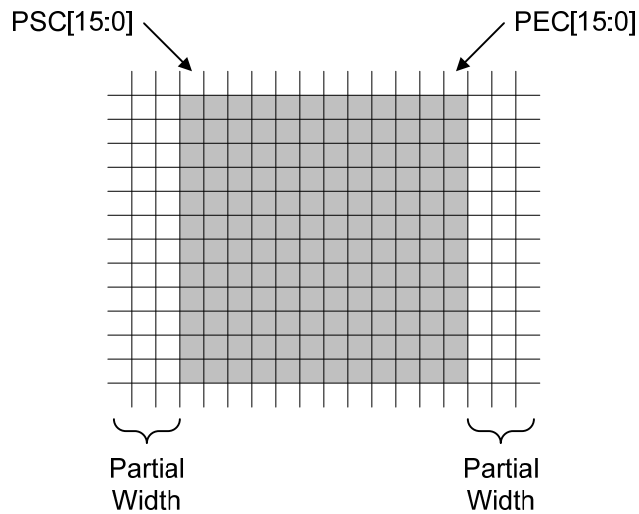
1307 If Start Column > End Column



1308

1309

Figure 59 set_partial_columns with set_address_mode B2 = 0



1310

1311

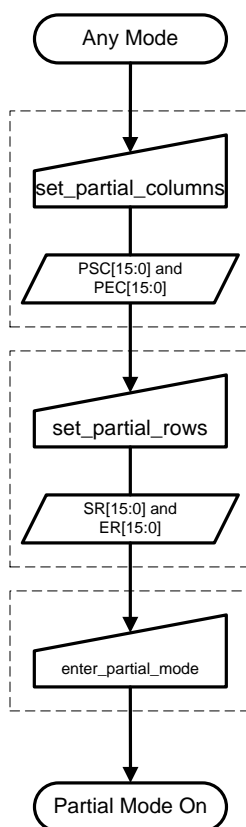
Figure 60 set_partial_columns with set_address_mode B2 = 1

1312 **Restrictions**

1313 PSC[15:0] and PEC[15:0] cannot be 0000h nor exceed the last horizontal column number.

1314 **Flow Chart**

1315 To enter Partial Display mode

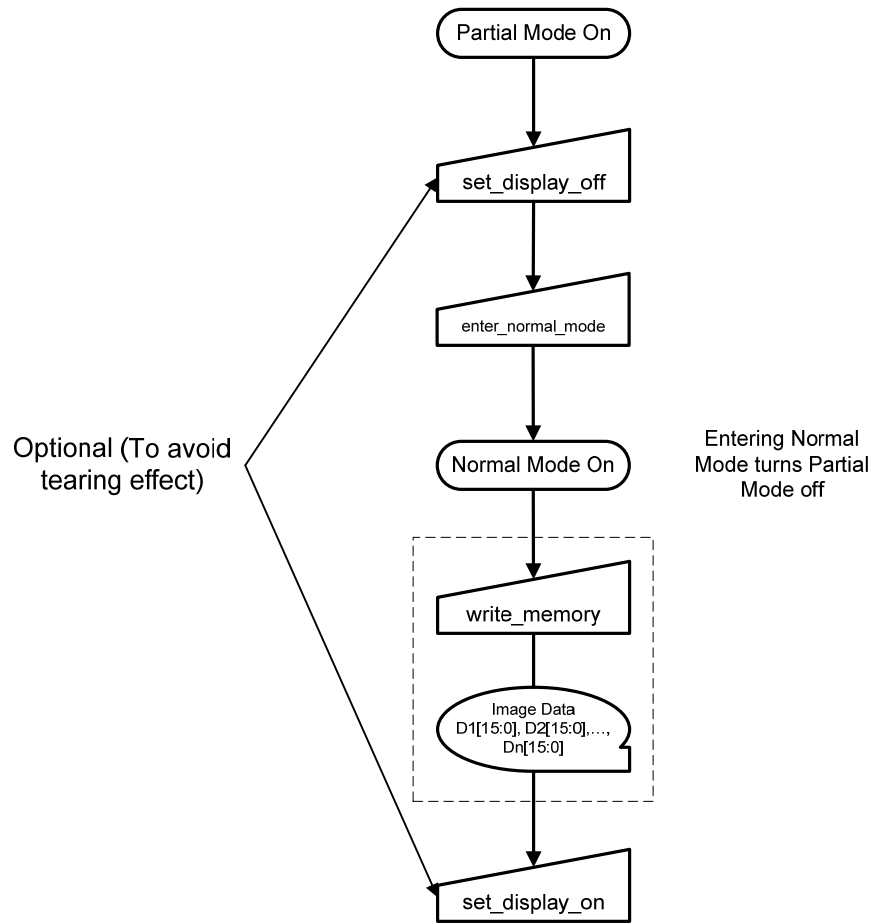


1316

1317

Figure 61 Entering Partial Display Mode Flow Chart

1318 To exit Partial Display mode



1319

1320

1321

Figure 62 Exiting Partial Display Mode Flow Chart

1322 **6.31 set_partial_rows**

1323 **Interface** All
 1324 **Command** 30h
 1325 **Parameters** See below

1326 **Command**

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

1327 **Parameter 1**

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

1328 **Parameter 2**

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

1329 **Parameter 3**

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

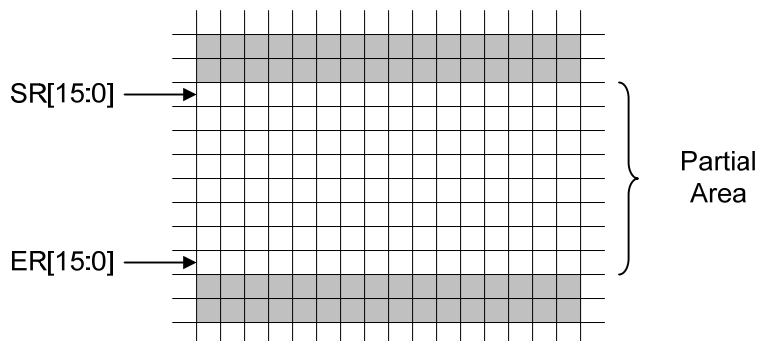
1330 **Parameter 4**

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

1331 **Description**

1332 This command defines the Partial Display mode's display height. There are two parameters associated with
 1333 this command, the first defines the Start Row (SR) and the second the End Row (ER), as illustrated
 1334 in Figure 63 through Figure 66. SR and ER refer to the Frame Memory Line Pointer.

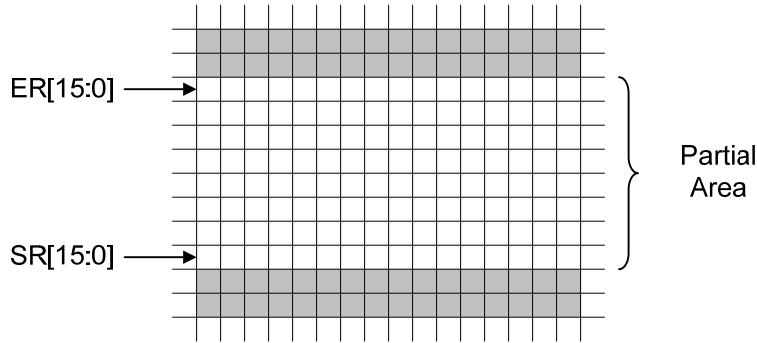
1335 If End Row > Start Row



1336

1337

Figure 63 set_partial_rows with set_address_mode B4 = 0

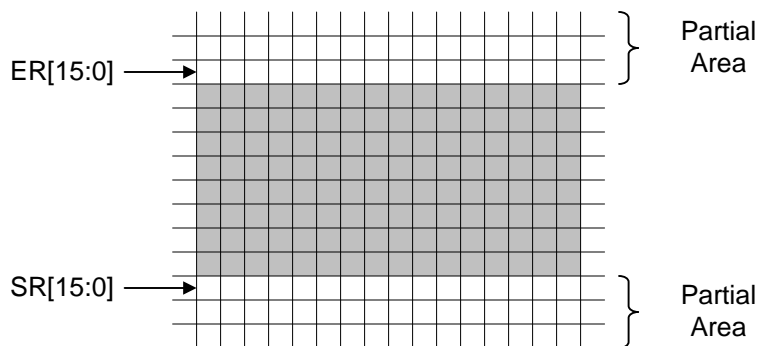


1338

1339

Figure 64 set_partial_rows with set_address_mode B4=1

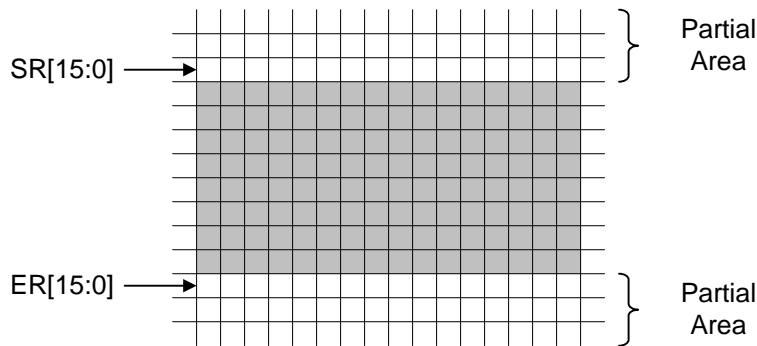
1340 If Start Row > End Row



1341

1342

Figure 65 set_partial_rows with set_address_mode B4 = 0



1343

1344

Figure 66 set_partial_rows with set_address_mode B4 = 1

1345 **Restrictions**

1346 SR[15:0] and ER[15:0] cannot be 0000h nor exceed the last vertical line number.

1347 **Flow Chart**

1348 See section 6.30.

1349

1350 **6.32 set_pixel_format**1351 **Interface** All1352 **Command** 3Ah1353 **Parameters** See below1354 **Command**

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

1355 **Parameter**

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

1356 **Description**

1357 This command sets the pixel format for the RGB image data used by the interface.

1358 Bits D[6:4] – DPI Pixel Format Definition

1359 Bits D[2:0] – DBI Pixel Format Definition

1360 Bits D7 and D3 are not used.

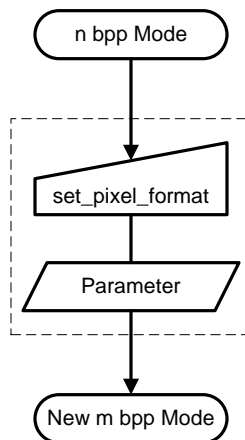
1361 The pixel formats are shown in Table 6.

1362 If a particular interface, either DBI or DPI, is not used then the corresponding bits in the parameter are ignored.
1363

1364 In 12, 16 & 18 bits/Pixel modes, the LUT is applied to transfer data into the frame memory.

1365 **Restrictions**

1366 There is no visible effect until the frame memory is written.

1367 **Flow Chart**

1368

1369

1370

Figure 67 set_pixel_format Flow Chart

1371 **6.33 set_scroll_area**

1372 **Interface** All
 1373 **Command** 33h
 1374 **Parameters** See below

1375 **Command**

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

1376 **Parameter 1**

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

1377 **Parameter 2**

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

1378 **Parameter 3**

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

1379 **Parameter 4**

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

1380 **Parameter 5**

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

1381 **Parameter 6**

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

1382 **Description**

1383 This command defines the display module's Vertical Scrolling Area.

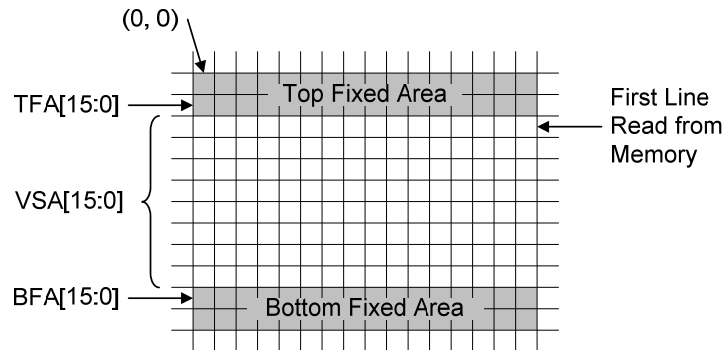
1384 If set_address_mode B4 = 0:

1385 The 1st & 2nd parameter, TFA[15:0], describes the Top Fixed Area in number of lines from the top of the
 1386 frame memory. The top of the frame memory and top of the display device are aligned.

1387 The 3rd & 4th parameter, VSA[15:0], describes the height of the Vertical Scrolling Area in number of lines
 1388 of frame memory from the Vertical Scrolling Start Address. The first line of the Vertical Scrolling Area
 1389 starts immediately after the bottom most line of the Top Fixed Area. The last line of the Vertical Scrolling
 1390 Area ends immediately before the top most line of the Bottom Fixed Area.

1391 The 5th & 6th parameter, BFA[15:0], describes the Bottom Fixed Area in number of lines from the bottom of
 1392 the frame memory. The bottom of the frame memory and bottom of the display device are aligned.

1393 TFA, VSA and BFA refer to the Frame Memory Line Pointer.



1394

1395

Figure 68 set_scroll_area set_address_mode B4 = 1 Example

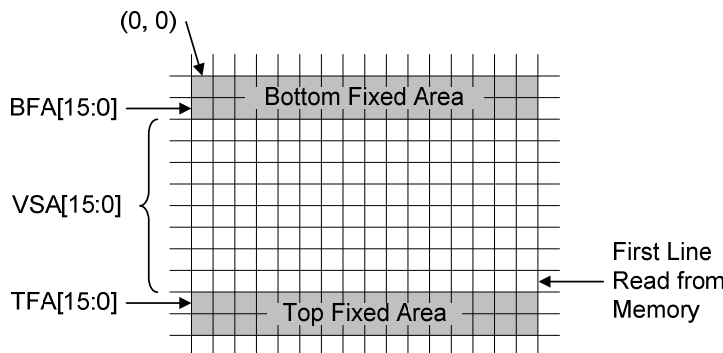
1396 If set_address_mode B4 = 1:

1397 The 1st & 2nd parameter, TFA[15:0], describes the Top Fixed Area in number of lines from the bottom of
 1398 the frame memory. The bottom of the frame memory and bottom of the display device are aligned.

1399 The 3rd & 4th parameter, VSA[15:0], describes the height of the Vertical Scrolling Area in number of lines
 1400 of frame memory from the Vertical Scrolling Start Address. The first line of the Vertical Scrolling Area
 1401 starts immediately after the top most line of the Top Fixed Area. The last line of the Vertical Scrolling Area
 1402 ends immediately before the bottom most line of the Bottom Fixed Area.

1403 The 5th & 6th parameter, BFA[15:0], describes the Bottom Fixed Area in number of lines from the top of the
 1404 frame memory. The top of the frame memory and top of the display device are aligned.

1405 TFA, VSA and BFA refer to the Frame Memory Line Pointer.



1406

1407

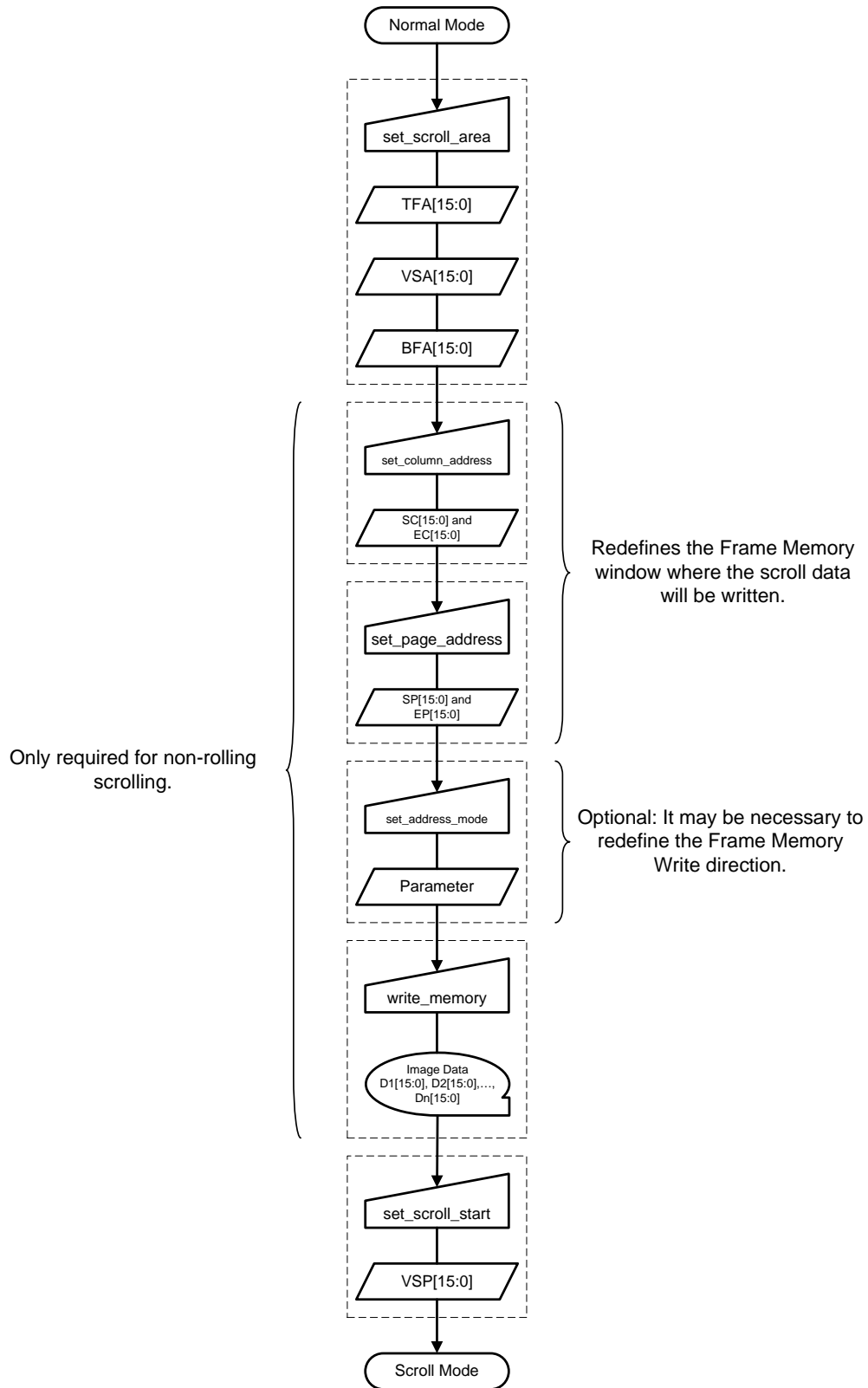
Figure 69 set_scroll_area set_address_mode B4 = 1 Example

1408 Restrictions

1409 The sum of TFA, VSA and BFA must equal the number of the display device's horizontal lines (pages),
 1410 otherwise Scrolling mode is undefined.

1411 In Vertical Scroll Mode, set_address_mode B5 should be set to '0' – this only affects the Frame Memory
 1412 Write.

1413 **Flow Chart**



1414

1415

Figure 70 set_scroll_area Flow Chart

1416 **6.34 set_scroll_start**
 1417 **Interface** All
 1418 **Command** 37h
 1419 **Parameters** See below

1420 **Command**

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

1421 **Parameter 1**

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

1422 **Parameter 2**

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

1423 **Description**

1424 This command sets the start of the vertical scrolling area in the frame memory. The vertical scrolling area is
 1425 fully defined when this command is used with the set_scroll_area command

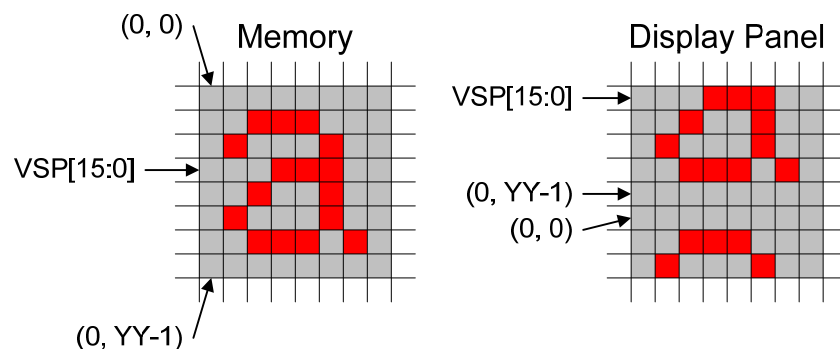
1426 The set_scroll_start command has one parameter, the Vertical Scroll Pointer. The VSP defines the line in
 1427 the frame memory that is written to the display device as the first line of the vertical scroll area. See
 1428 section 6.33 for a description of the vertical scroll area.

1429 The displayed image also depends on the setting of the Line Address Order bit, B4, in the
 1430 set_address_mode register. See the examples below.

1431 If set_address_mode B4 = 0:

1432 Example:

1433 When Top Fixed Area = Bottom Fixed Area = 0, Vertical Scrolling Area = YY and VSP = 3.



1434

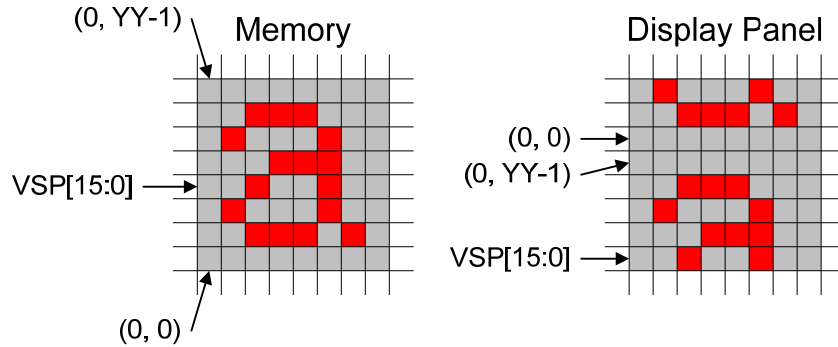
1435

Figure 71 set_scroll_start set_address_mode B4 = 0

1436 If set_address_mode B4 = 1:

1437 Example:

1438 When Top Fixed Area = Bottom Fixed Area = 0, Vertical Scrolling Area = YY and VSP = 3.



1439

1440

Figure 72 set_scroll_start set_address_mode B4 = 1

1441 Restrictions

1442 Since the value of the Vertical Scrolling Start Address is absolute with reference to the Frame Memory, it
 1443 must not enter the fixed areas, see section 6.33, otherwise an undesirable image may be shown on the
 1444 Display Panel.

1445 The following conditions shall apply:

1446 If set_address_mode B4 = 0, $TFA[15:0] - 1 < VSP[15:0] < \# \text{ of lines in frame memory} - BFA[15:0]$

1447 If set_address_mode B4 = 1, $BFA[15:0] - 1 < VSP[15:0] < \# \text{ of lines in frame memory} - TFA[15:0]$

1448 Flow Chart

1449 See section 6.33 description.

1450

1451 **6.35 set_tear_off**
 1452 **Interface** All
 1453 **Command** 34h
 1454 **Parameters** None

1455 **Command**

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

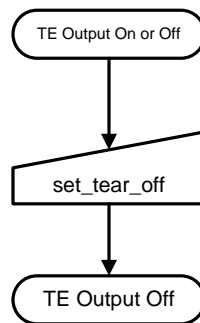
1456 **Description**

1457 This command turns off the display module's Tearing Effect output signal on the TE signal line.

1458 **Restrictions**

1459 This command has no effect when the Tearing Effect output is already off.

1460 **Flow Chart**



1461

1462

1463

Figure 73 set_tear_off Flow Chart

1464 **6.36 set_tear_on**
 1465 **Interface** All
 1466 **Command** 35h
 1467 **Parameters** See below

1468 **Command**

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

1469 **Parameter**

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

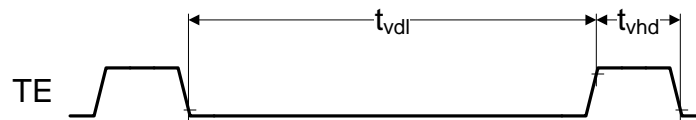
1470 **Description**

1471 This command turns on the display module's Tearing Effect output signal on the TE signal line. The TE
 1472 signal is not affected by changing set_address_mode bit B4.

1473 set_tear_on has one parameter that describes the Tearing Effect Output Line mode.

1474 If M = 0 (Mode 0):

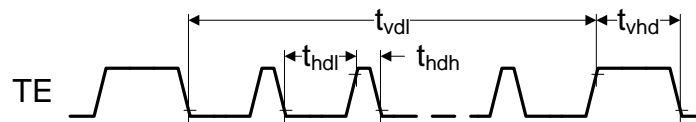
1475 The Tearing Effect Output line consists of V-Blanking information only.



1476
 1477 **Figure 74 set_tear_on M = 0**

1478 If M = 1 (Mode 1):

1479 The Tearing Effect Output Line consists of both V-Blanking and H-Blanking information.



1480
 1481 **Figure 75 set_tear_on M = 1**

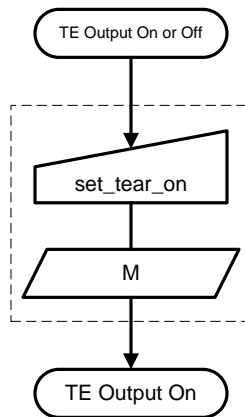
1482 The Tearing Effect Output line shall be active low when the display module is in Sleep mode.

1483 See [MIPI02] for definitions of t_{vdl} , t_{vhd} , t_{hdl} and t_{hdh} .

1484 **Restrictions**

1485 This command takes affect on the frame following the current frame. Therefore, if the Tear Effect (TE)
 1486 output is already ON, the TE output shall continue to operate as programmed by the previous set_tear_on,
 1487 or set_tear_scanline, command until the end of the frame.

1488 **Flow Chart**



1489

1490

1491

Figure 76 set_tear_on Flow Chart

1492 **6.37 set_tear_scanline**

1493 **Interface** All
 1494 **Command** 44h
 1495 **Parameters** See below

1496 **Command**

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

1497 **Parameter 1**

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

1498 **Parameter 2**

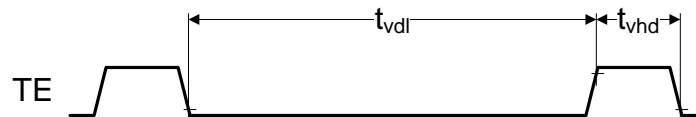
| Direction | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Hex Code |
|-----------|----|----|----|----|----|----|----|----|----------|
| H→D | N7 | N6 | N5 | N4 | N3 | N2 | N1 | N0 | XXh |

1499 **Description**

1500 This command turns on the display module's Tearing Effect output signal on the TE signal line when the
 1501 display module reaches line N. The TE signal is not affected by changing set_address_mode bit B4.

1502 The Tearing Effect Line On has one parameter that describes the Tearing Effect Output Line mode.

1503 After issuing a set_tear_scanline command to the display module, the Tearing Effect output signal, e.g. as
 1504 in DBI-2 systems, shall be a delayed version of V-Blanking information as illustrated by Figure 77.



1505

1506

Figure 77 set_tear_scanline

1507 Note that set_tear_scanline with $N = 0$ is equivalent to set_tear_on with $M = 0$.

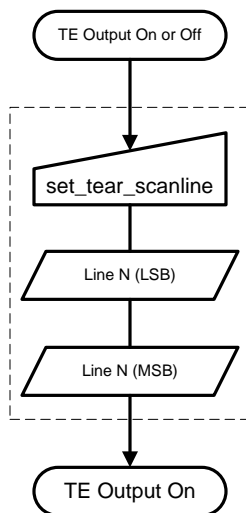
1508 The Tearing Effect Output line shall be active low when the display module is in Sleep mode.

1509 See [MIPI02] for definitions of t_{vdl} and t_{vhd} and [MIPI03] for definition of display module line numbers.

1510 **Restrictions**

1511 This command takes affect on the frame following the current frame. Therefore, if the Tear Effect (TE)
 1512 output is already ON, the TE output shall continue to operate as programmed by the previous set_tear_on,
 1513 or set_tear_scanline, command until the end of the frame.

1514 **Flow Chart**



1515

1516

1517

Figure 78 set_tear_scanline Flow Chart

1518 **6.38 soft_reset**
 1519 **Interface** All
 1520 **Command** 01h
 1521 **Parameters** None

1522 **Command**

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

1523 **Description**

1524 The display module performs a software reset. Registers are written with their SW Reset default values.
 1525 See section 5.7 for a list of the reset values.

1526 Frame Memory contents are unaffected by this command.

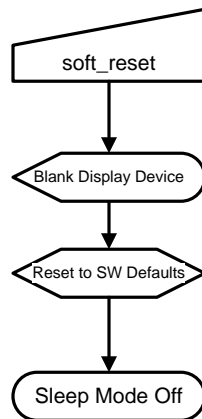
1527 **Restrictions**

1528 The host processor must wait five milliseconds before sending any new commands to a display module
 1529 following this command. The display module updates the registers during this time.

1530 If a soft_reset is sent when the display module is in Sleep Mode, the host processor must wait 120
 1531 milliseconds before sending an exit_sleep_mode command.

1532 soft_reset should not be sent when the display module is not in Sleep mode.

1533 **Flow Chart**



1534

1535

Figure 79 soft_reset Flow Chart

1536

1537 **6.39 write_LUT**

1538 **Interface** All
 1539 **Command** 2Dh
 1540 **Parameters** See below

1541 **Command**

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

1542 **Parameter 1**

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

1543 .
 1544 .
 1545 .

1546 **Parameter N**

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

1547 **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 |

1548 .
 1549 .
 1550 .

1551 **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 |

1552 **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 |

1553 .
 1554 .
 1555 .

1556 **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 |

1557 **Description**

1558 This command sets the LUT for pixel color depth conversions. Six conversions are supported as indicated
 1559 in Table 8.

1560

Table 8 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 |

1561 The LUT size depends on the pixel format of the display module. In the list below, N is the number of red
 1562 or blue components and M is the number of green components in the LUT.

1563 16-bit color display modules: $N = M = 16$; Total LUT Size = $2*N + M = 48$ bytes.

1564 18-bit color display modules: $N = 32, M = 64$; Total LUT Size = $2*N + M = 128$ bytes.

1565 24-bit color display modules: $N = M = 64$; Total LUT Size = $2*N + M = 192$ bytes.

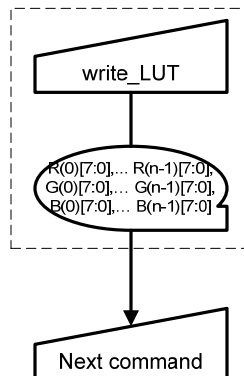
1566 Regardless of host processor color depth, the defined size of the LUT shall be written according to the
 1567 number of colors supported by the display module. See Annex A.

1568 This command has no effect on other commands or the contents of frame memory. Visible changes take
 1569 effect the next time the frame memory is written.

1570 **Restrictions**

1571 None

1572 **Flow Chart**



1573

1574

1575

Figure 80 write_LUT Flow Chart

1576 **6.40 write_memory_continue**

1577 **Interface** All
 1578 **Command** 3Ch
 1579 **Parameters** See below

1580 **Command**

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

1581 **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 |

1582

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

1583

1584

1585

.
.
.

1586 **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 |

1587

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

1588 **Description**

1589 This command transfers image data from the host processor to the display module's frame memory
 1590 continuing from the pixel location following the previous write_memory_continue or write_memory_start
 1591 command.

1592 If set_address_mode B5 = 0:

1593 Data is written continuing from the pixel location after the write range of the previous write_memory_start
 1594 or write_memory_continue. The column register is then incremented and pixels are written to the frame
 1595 memory until the column register equals the End Column (EC) value. The column register is then reset to
 1596 SC and the page register is incremented. Pixels are written to the frame memory until the page register
 1597 equals the End Page (EP) value and the column register equals the EC value, or the host processor sends
 1598 another command. If the number of pixels exceeds $(EC - SC + 1) * (EP - SP + 1)$ the extra pixels are
 1599 ignored.

1600 If set_address_mode B5 = 1:

1601 Data is written continuing from the pixel location after the write range of the previous write_memory_start
 1602 or write_memory_continue. The page register is then incremented and pixels are written to the frame
 1603 memory until the page register equals the End Page (EP) value. The page register is then reset to SP and the
 1604 column register is incremented. Pixels are written to the frame memory until the column register equals the

1605 End column (EC) value and the page register equals the EP value, or the host processor sends another
 1606 command. If the number of pixels exceeds $(EC - SC + 1) * (EP - SP + 1)$ the extra pixels are ignored.

1607 See section 6.25 for descriptions of the Start Column and End Column values.

1608 See section 6.29 for descriptions of the Start Page and End Page values.

1609 See section 8 in [MIPI01] and section 10 in [MIPI02] for color encoding for 8 or 9 data bit image data.

1610 Note the command description above shows 16-bit pixel data transferred over a 16-bit bus. Other
 1611 possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel
 1612 data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.

1613 The relationship between some common colors and the corresponding image data are shown in the
 1614 following table.

1615

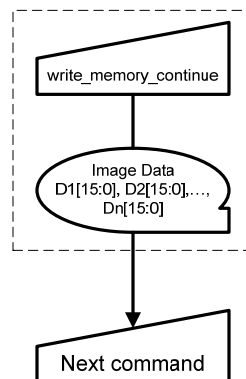
Table 9 Common Color Encoding

| 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 |

1616 **Restrictions**

1617 A write_memory_start should follow a set_column_address, set_page_address or set_address_mode to
 1618 define the write address. Otherwise, data written with write_memory_continue is written to undefined
 1619 addresses.

1620 **Flow Chart**



1621

1622

Figure 81 write_memory_continue Flow Chart

1623

1624 **6.41 write_memory_start**

1625 **Interface** All
 1626 **Command** 2Ch
 1627 **Parameters** See below

1628 **Command**

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

1629 **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 |

1630

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

1631

1632

1633

.
.
.

1634 **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 |

1635

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

1636 **Description**

1637 This command transfers image data from the host processor to the display module's frame memory starting
 1638 at the pixel location specified by preceding set_column_address and set_page_address commands.

1639 If set_address_mode B5 = 0:

1640 The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

1641 Pixel Data 1 is stored in frame memory at (SC, SP). The column register is then incremented and pixels are
 1642 written to the frame memory until the column register equals the End Column (EC) value. The column
 1643 register is then reset to SC and the page register is incremented. Pixels are written to the frame memory
 1644 until the page register equals the End Page (EP) value and the column register equals the EC value, or the
 1645 host processor sends another command. If the number of pixels exceeds $(EC - SC + 1) * (EP - SP + 1)$ the
 1646 extra pixels are ignored.

1647 If set_address_mode B5 = 1:

1648 The column and page registers are reset to the Start Column (SC) and Start Page (SP), respectively.

1649 Pixel Data 1 is stored in frame memory at (SC, SP). The page register is then incremented and pixels are
 1650 written to the frame memory until the page register equals the End Page (EP) value. The page register is
 1651 then reset to SP and the column register is incremented. Pixels are written to the frame memory until the

1652 column register equals the End column (EC) value and the page register equals the EP value, or the host
 1653 processor sends another command. If the number of pixels exceeds $(EC - SC + 1) * (EP - SP + 1)$ the extra
 1654 pixels are ignored.

1655 See section 6.25 for descriptions of the Start Column and End Column values.

1656 See section 6.29 for descriptions of the Start Page and End Page values.

1657 See section 8 in [MIPI01] and section 10 in [MIPI02] for color encoding for 8 or 9 data bit image data.

1658 Note the command description above shows 16-bit pixel data transferred over a 16-bit bus. Other
 1659 possibilities not illustrated here would show 9-bit pixel data transferred over a 9-bit bus, and 8-bit pixel
 1660 data transferred over an 8-bit bus. See Annex A for details on pixel to byte mapping.

1661 The relationship between some common colors and the corresponding image data are shown in the
 1662 following table.

1663

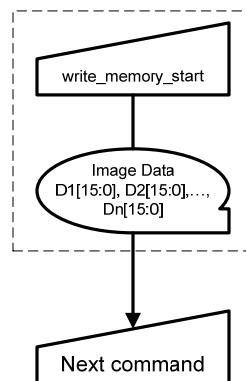
Table 10 Common Color Encoding

| 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 |

1664 **Restrictions**

1665 A write_memory_start should follow a set_column_address, set_page_address or set_address_mode to
 1666 define the write location. Otherwise, data written with write_memory_start and any following
 1667 write_memory_continue commands is written to undefined locations.

1668 **Flow Chart**



1669

1670

Figure 82 write_memory_start Flow Chart

1671

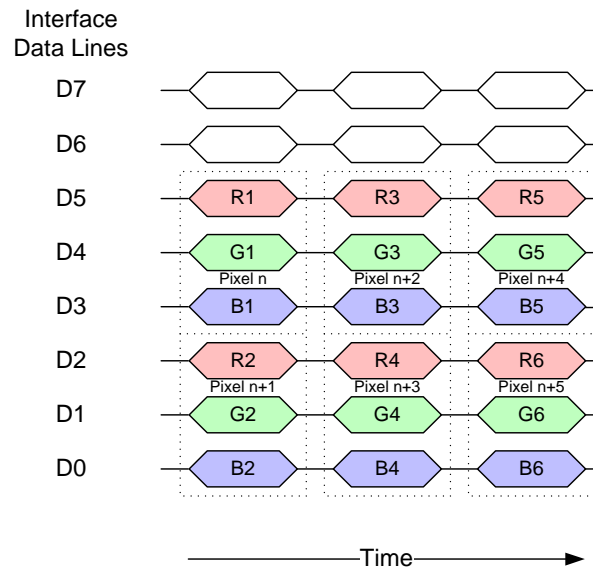
1672 Annex A Pixel-to-Byte Mapping

1673 Many of the commands in this specification utilize display panel properties and therefore refer to pixels and
 1674 scan lines. However, numerous components of a display system are inherently byte oriented. Therefore, a
 1675 consistent method should be used to convert pixel formats to bytes to ensure interoperability among all
 1676 components. This section defines the pixel-to-byte mapping used by this specification.

1677 Note the `set_address_mode` command (section 6.24) affects the bit ordering within a pixel, red and blue
 1678 components may be swapped, and the order pixels are transferred across the interface. The figures in this
 1679 section are shown with `set_address_mode B4=B5=B6=B7=0`.

1680 A.1 Three Bits per Pixel Format

1681 Three bits per pixel formats do not map directly to byte boundaries and therefore require special handling.
 1682 In this pixel format, each byte holds two pixels. Two bits in each byte convey no color information. The
 1683 organization of bits is shown in Figure 83.



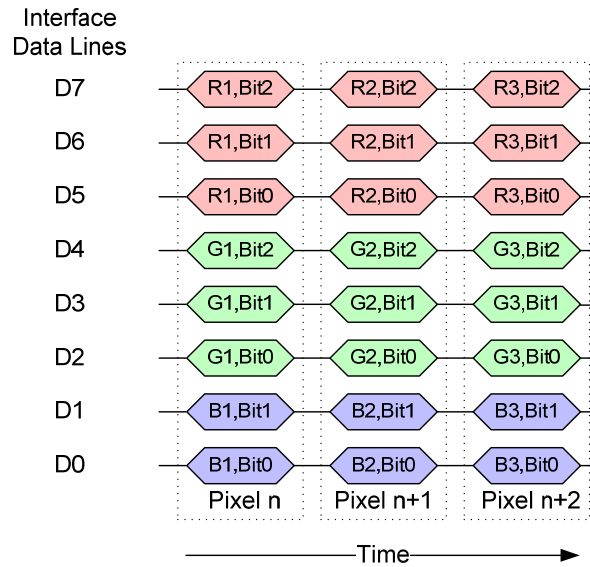
1684

1685

Figure 83 Three Bits per Pixel Format to Byte Mapping

1686 **A.2 Eight Bits per Pixel Format**

1687 Eight bits per pixel formats map directly to byte boundaries and therefore require no special
 1688 handling. Figure 84 shows the mapping of pixels to bytes.



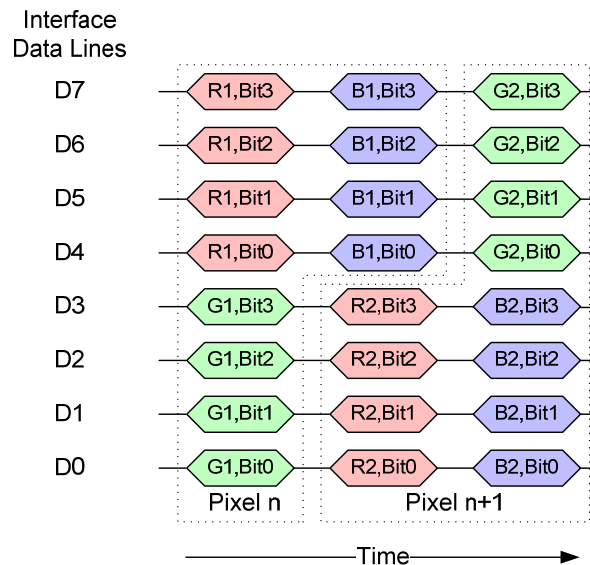
1689

1690

Figure 84 Eight Bits per Pixel Format to Byte Mapping

1691 **A.3 Twelve Bits per Pixel Format**

1692 Twelve bits per pixel formats do not map directly to byte boundaries and therefore require special handling.
 1693 In this pixel format, three bytes hold two pixels. Figure 85 shows the mapping of pixels to bytes.



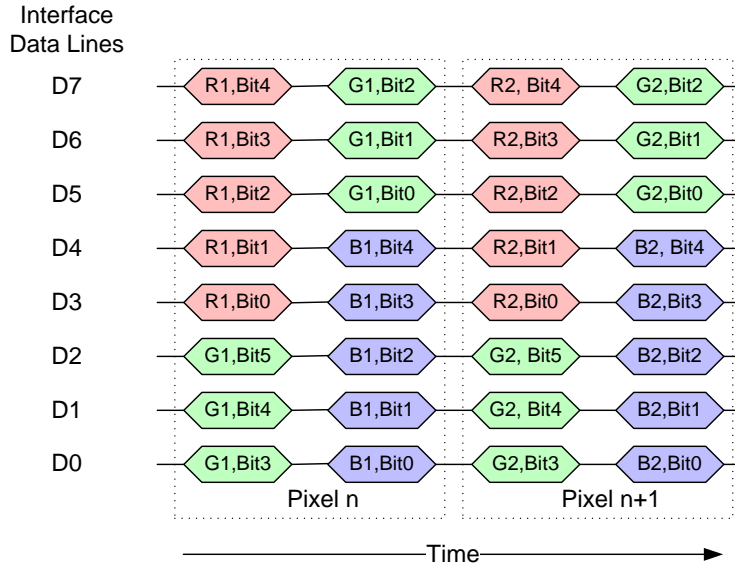
1694

1695

Figure 85 Twelve Bits per Pixel Format to Byte Mapping

1696 **A.4 Sixteen Bits per Pixel Format**

1697 Sixteen bits per pixel formats do not map directly to byte boundaries and therefore require special handling.
 1698 However, this format is simpler than twelve bit formats since one pixel occupies two bytes. Figure 86
 1699 shows the mapping of pixels to bytes.



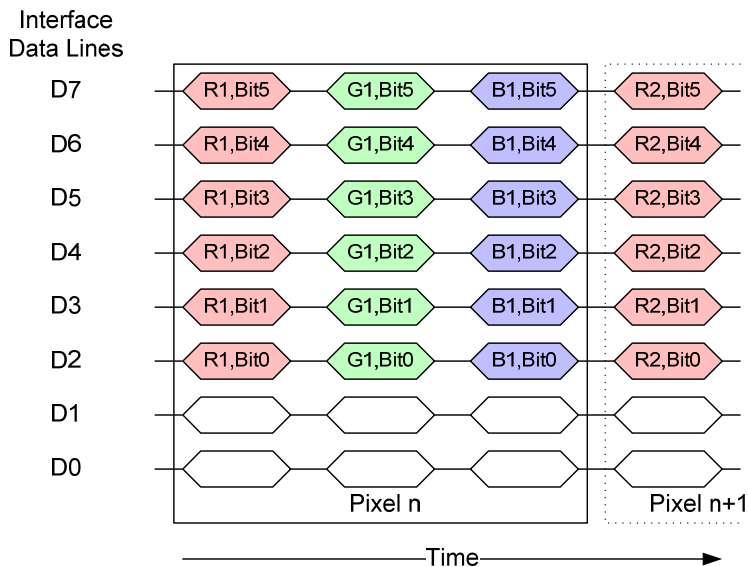
1700

1701

Figure 86 Sixteen Bits per Pixel Format to Byte Mapping

1702 **A.5 Eighteen Bits per Pixel Format**

1703 Eighteen bits per pixel formats do not map directly to byte boundaries and therefore require special
 1704 handling. In this pixel format, each pixel occupies three bytes (24-bits), one for each color component. Two
 1705 bits in each byte convey no color information. Figure 87 shows the mapping of pixels to bytes.



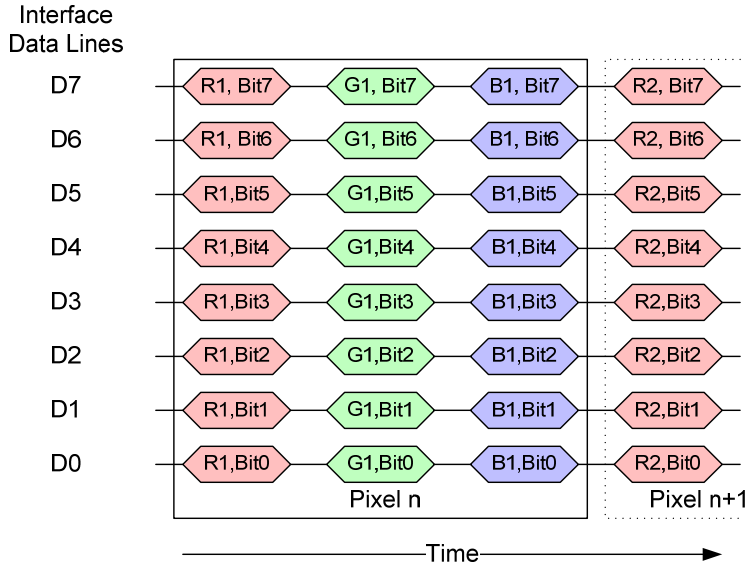
1706

1707

Figure 87 Eighteen Bits per Pixel Format to Byte Mapping

1708 **A.6 Twenty-four Bits per Pixel Format**

1709 Twenty-four bits per pixel formats do not map directly to byte boundaries and therefore require special
 1710 handling. This format is similar to the eighteen bits per pixel format since one pixel occupies three bytes.
 1711 However, all bits in this format convey color information. Figure 88 shows the mapping of pixels to bytes.



1712

1713

Figure 88 Twenty-four Bits per Pixel Format to Byte Mapping

1714

1715 **Annex B Color Depth Conversion Look-up Tables**
 1716 **(informative)**

1717 **B.1 Color Depth Conversion LUT – 12-bit Color to 16-bit Color**

1718 **Table 11 12-bit to 16-bit LUT Red Component Values**

| R input (4-bit) 12-bits/pixel 4,096 colors | R output (5-bit) 16-bits/pixel 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 |

1719

1720

Table 12 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 |

1721

1722

Table 13 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 |

1723

1724
1725**B.2 Color Depth Conversion LUT – 12-bit and 16-bit Colors to 18-bit Color****Table 14 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 |

| 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 |
|---|---|---|--------------------------------|
| 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 |

1726

1727

Table 15 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 |

| 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 | 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 |
| 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 ₄₅₄ 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 |

1729

Table 16 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 |

1730

1731
1732
1733

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

Table 17 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 |

| 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 | 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 |
| 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 |

| 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 | 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 |

1734

1735

Table 18 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 |

| 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 | 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 |
| 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 |

| 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 | 111111 | 111111 | $G_{637} G_{636} G_{635} G_{634} G_{633} G_{632} G_{631} G_{630}$ | 128 |

1736

1737

Table 19 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 |

| 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 | 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 |
| 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 |

| 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 | 111111 | B ₆₃₇ B ₆₃₆ B ₆₃₅ B ₆₃₄ B ₆₃₃ B ₆₃₂ B ₆₃₁ B ₆₃₀ | 192 |

1738