



DesignWare® Cores MIPI DSI Host Controller

Databook

DWC MIPI DSI HOST CONTROLLER - Product Code: 7610-0
MIPI DSI HOST COMBO CONTROLLER - Product Code: B759-0
MIPI DSI HOST CONTROLLER with VESA DSC ENCODER - Product Code: C640-0
MIPI DSI HOST COMBO CONTROLLER with VESA DSC ENCODER - Product Code: C641-0

Synopsys confidential documentation
under use and disclosure restricted
Axera, not redistribute.

Copyright Notice and Proprietary Information

© 2021 Synopsys, Inc. All rights reserved. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at
<https://www.synopsys.com/company/legal/trademarks-brands.html>

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.

www.synopsys.com

Contents

Revision History	9
Preface	15
Databook Organization	15
Related Documentation	15
Web Resources	16
Synopsys Statement on Inclusivity and Diversity	16
Customer Support	16
Chapter 1	
Product Overview	19
1.1 General Product Description	20
1.1.1 Standards Compliance	20
1.1.2 Supported Features	21
1.1.3 Unsupported Features	23
1.1.4 Applications	23
1.1.5 Requirements	24
1.1.6 Limitations and Exceptions	25
1.1.7 Resolution and Frame Rates	26
1.2 Related Synopsys Products	35
Chapter 2	
Architecture	37
2.1 Overview of Architecture	38
2.2 Selection of Appropriate Interface	40
2.3 DPI	41
2.3.1 Overview of DPI	41
2.3.2 Enabling DPI	41
2.3.3 Configuring the Display Pixel Interface	43
2.3.4 Video Mode with Stereoscopic Image Data	43
2.3.5 Video Transmission Modes	44
2.3.6 Updating DPI Configuration Dynamically	45
2.3.7 Signals Related to DPI	48
2.3.8 Registers Related to DPI	49
2.4 eDPI	50

2.4.1 Overview of eDPI	50
2.4.2 Enabling eDPI	50
2.4.3 Description of eDPI	50
2.4.4 eDPI in Adapted Command Mode	53
2.4.5 Support for Tearing Effect (TE)	54
2.4.6 Halt Functionality in Adapted Command Mode	56
2.4.7 Auto ULPS Control Scheme	61
2.4.8 Signals Related to eDPI	63
2.4.9 Registers Related to eDPI	63
2.5 DBI	64
2.5.1 Overview of DBI	64
2.5.2 Enabling DBI	64
2.5.3 DSI Packet Types	64
2.5.4 Configuring the DBI Pixel Data	65
2.5.5 Partitioning the Long write_memory_start Commands	65
2.5.6 Handling the Low Transmission Speed	67
2.5.7 Dimensioning the Size of FIFOs for Automatic Partitioning	68
2.5.8 Signals Related to DBI	69
2.5.9 Registers Related to DBI	69
2.6 Continuous Clock Using C-PHY Mode	70
2.7 APB Slave Generic Interface	71
2.7.1 Overview of APB Slave Generic Interface	71
2.7.2 Description of APB Slave Generic Interface	71
2.7.3 Packet Transmission Using the Generic Interface	72
2.7.4 Signals Related to APB Slave Generic Interface	75
2.7.5 Registers Related to APB Slave Generic Interface	75
2.8 Transmission of Commands	76
2.8.1 Transmission of Commands in Video Mode	76
2.8.2 Transmission of Commands in Low-Power Mode	78
2.8.3 Transmission of Commands in High-Speed Mode	82
2.8.4 Read Command Transmission	82
2.8.5 Clock Lane in Low-Power Mode (D-PHY)	83
2.9 Virtual Channels	85
2.9.1 Overview of Virtual Channels	85
2.9.2 Registers Related to Virtual Channels	86
2.10 Video Mode Pattern Generator	87
2.10.1 Overview of Video Mode Pattern Generator	87
2.10.2 Registers Related to Video Mode Pattern Generator	87
2.10.3 Color Bar Pattern	88
2.10.4 Color Coding	88
2.10.5 BER Testing Pattern	90
2.10.6 Video Mode Pattern Generator Resolution	90
2.11 Display Stream Compression	92
2.11.1 DSC Rate Buffer RAM Requirements	94
2.11.2 Enabling Compression	95
2.11.3 DSC 24 - External DSC Encoder	96
2.11.4 DSC Encoder - Internal Compression	97
2.12 Supported 2-Port RAM Types	99
2.13 Timeout Timers	100

2.13.1 Peripheral Response Timeout	100
2.13.2 Errors Raised by Timeouts	103
2.14 Error Control	105

Chapter 3

Parameter Descriptions	107
3.1 Core Configuration Parameters	108
3.2 FIFOs Configuration Parameters	110
3.3 VESA DSC Configuration Parameters	115

Chapter 4

Signal Descriptions	121
4.1 AMBA APB Signals	123
4.2 DSI Host Generic Signals	126
4.3 PHY PPI Interface Signals	127
4.4 DPI/eDPI Interface Signals	139
4.5 DBI Interface to 2-Port RAM Signals	144
4.6 DBI Interface Shared Signals	150
4.7 DBI Type A Clocked E Interface Signals	152
4.8 DBI Type B Interface Signals	153
4.9 DBI Write Interface to 2-Port RAM Signals	154
4.10 DBI Read Interface to 2-Port RAM Signals	156
4.11 Generic Write Interface to 2-Port RAM Signals	158
4.12 Generic Read Interface to 2-Port RAM Signals	161
4.13 DSI Host Generic Signals	164
4.14 Vendor Specific Interface Signals	165
4.15 PHY External Interface Signals	167
4.16 DSC Encoder Memory Interface Signals	168

Chapter 5

Register Descriptions	181
5.1 DSI Registers	189
5.1.1 VERSION	190
5.1.2 PWR_UP	191
5.1.3 CLKMGR_CFG	192
5.1.4 DPI_VCID	193
5.1.5 DPI_COLOR_CODING	194
5.1.6 DPI_CFG_POL	196
5.1.7 DPI_LP_CMD_TIM	198
5.1.8 DBI_VCID	199
5.1.9 DBI_CFG	200
5.1.10 DBI_PARTITIONING_EN	202
5.1.11 DBI_CMDSIZE	203
5.1.12 PCKHDL_CFG	204
5.1.13 GEN_VCID	206
5.1.14 MODE_CFG	207
5.1.15 VID_MODE_CFG	208

5.1.16 VID_PKT_SIZE	211
5.1.17 VID_NUM_CHUNKS	212
5.1.18 VID_NULL_SIZE	213
5.1.19 VID_HSA_TIME	214
5.1.20 VID_HBP_TIME	215
5.1.21 VID_HLINE_TIME	216
5.1.22 VID_VSA_LINES	217
5.1.23 VID_VBP_LINES	218
5.1.24 VID_VFP_LINES	219
5.1.25 VID_VACTIVE_LINES	220
5.1.26 EDPI_CMD_SIZE	221
5.1.27 CMD_MODE_CFG	222
5.1.28 GEN_HDR	226
5.1.29 GEN_PLD_DATA	227
5.1.30 CMD_PKT_STATUS	228
5.1.31 TO_CNT_CFG	232
5.1.32 HS_RD_TO_CNT	234
5.1.33 LP_RD_TO_CNT	235
5.1.34 HS_WR_TO_CNT	236
5.1.35 LP_WR_TO_CNT	237
5.1.36 BTA_TO_CNT	238
5.1.37 SDF_3D	239
5.1.38 LPCLK_CTRL	241
5.1.39 PHY_TMR_LPCLK_CFG	242
5.1.40 PHY_TMR_CFG	243
5.1.41 PHY_RSTZ	244
5.1.42 PHY_IF_CFG	245
5.1.43 PHY_ULPS_CTRL	246
5.1.44 PHY_TX_TRIGGER	247
5.1.45 PHY_STATUS	248
5.1.46 PHY_TST_CTRL0	251
5.1.47 PHY_TST_CTRL1	252
5.1.48 INT_ST0	253
5.1.49 INT_ST1	256
5.1.50 INT_MSK0	259
5.1.51 INT_MSK1	262
5.1.52 PHY_CAL	265
5.1.53 INT_FORCE0	266
5.1.54 INT_FORCE1	269
5.1.55 AUTO_ULPS_MODE	272
5.1.56 AUTO_ULPS_ENTRY_DELAY	273
5.1.57 AUTO_ULPS_WAKEUP_TIME	274
5.1.58 DSC_PARAMETER	275
5.1.59 PHY_TMR_RD_CFG	277
5.1.60 AUTO_ULPS_MIN_TIME	278
5.1.61 PHY_MODE	279
5.1.62 VID_SHADOW_CTRL	280
5.1.63 DPI_VCID_ACT	281
5.1.64 DPI_COLOR_CODING_ACT	282

5.1.65	DPI_LP_CMD_TIM_ACT	284
5.1.66	EDPI_TE_HW_CFG	285
5.1.67	VID_MODE_CFG_ACT	287
5.1.68	VID_PKT_SIZE_ACT	289
5.1.69	VID_NUM_CHUNKS_ACT	290
5.1.70	VID_NULL_SIZE_ACT	291
5.1.71	VID_HSA_TIME_ACT	292
5.1.72	VID_HBP_TIME_ACT	293
5.1.73	VID_HLINE_TIME_ACT	294
5.1.74	VID_VSA_LINES_ACT	295
5.1.75	VID_VBP_LINES_ACT	296
5.1.76	VID_VFP_LINES_ACT	297
5.1.77	VID_VACTIVE_LINES_ACT	298
5.1.78	VID_PKT_STATUS	299
5.1.79	SDF_3D_ACT	302
5.1.80	DSC_ENC_COREID	304
5.1.81	DSC_ENC_VERSION	305
5.1.82	DSC_ENC_FLATNESS_DET_THRES	306
5.1.83	DSC_ENC_DELAY	307
5.1.84	DSC_ENC_COMPRESSED_LINE_SIZE	308
5.1.85	DSC_ENC_LINES_IN_EXCESS	309
5.1.86	DSC_ENC_RBUF_ADDR_LAST_LINE_ADJ	310
5.1.87	DSC_MODE	311
5.1.88	DSC_ENC_INT_ST	312
5.1.89	DSC_ENC_INT_MSK	315
5.1.90	DSC_ENC_INT_FORCE	318
5.1.91	DSC_FIFO_STATUS_SELECT	321
5.1.92	DSC_FIFO_STATUS	322
5.1.93	DSC_FIFO_STATUS2	324
5.1.94	DSC_FIFO_WORD_COUNT	326
5.1.95	DSC_FIFO_WORD_COUNT2	327
5.1.96	DSC_ENC_PPS_0_3	328
5.1.97	DSC_ENC_PPS_4_7	330
5.1.98	DSC_ENC_PPS_8_11	332
5.1.99	DSC_ENC_PPS_12_15	333
5.1.100	DSC_ENC_PPS_16_19	334
5.1.101	DSC_ENC_PPS_20_23	335
5.1.102	DSC_ENC_PPS_24_27	336
5.1.103	DSC_ENC_PPS_28_31	337
5.1.104	DSC_ENC_PPS_32_35	339
5.1.105	DSC_ENC_PPS_36_39	340
5.1.106	DSC_ENC_PPS_40_43	341
5.1.107	DSC_ENC_PPS_44_47	343
5.1.108	DSC_ENC_PPS_48_51	345
5.1.109	DSC_ENC_PPS_52_55	347
5.1.110	DSC_ENC_PPS_56_59	349
5.1.111	DSC_ENC_PPS_60_63	351
5.1.112	DSC_ENC_PPS_64_67	353
5.1.113	DSC_ENC_PPS_68_71	355

5.1.114 DSC_ENC_PPS_72_75	357
5.1.115 DSC_ENC_PPS_76_79	359
5.1.116 DSC_ENC_PPS_80_83	361
5.1.117 DSC_ENC_PPS_84_87	363
5.1.118 INT_ST0_AP	365
5.1.119 INT_MSK0_AP	368
5.1.120 INT_FORCE0_AP	371
5.1.121 INT_ST1_AP	374
5.1.122 INT_MSK1_AP	377
5.1.123 INT_FORCE1_AP	379
5.1.124 INT_ST2_AP	381
5.1.125 INT_MSK2_AP	384
5.1.126 INT_FORCE2_AP	387
5.1.127 TO_HSTXRDY_CFG_AP	390
5.1.128 TO_LPTXRDY_CFG_AP	391
5.1.129 TO_LPTXTRIG_CFG_AP	392
5.1.130 TO_LPTXULPS_CFG_AP	393
5.1.131 TO_HSTX_CFG_AP	394
5.1.132 TO_LPRX_CFG_AP	395
5.1.133 TO_BTA_CFG_AP	396
5.1.134 TO_CLK_DIV_AP	397
5.1.135 ERR_INJ_CTRL_AP	398
5.1.136 ERR_INJ_CHK_MSK_AP	399
5.1.137 ERR_INJ_DATA_MSK_AP	400
5.1.138 ERR_INJ_ST_AP	401
Appendix A Area and Power	403
Appendix B DBI Color Code Mapping Waveforms	415
Appendix C Error Handling	429
C.1 Causes of Errors and Recovery	430
Appendix D Static and Dynamic Registers	439
Appendix E Internal Parameter Descriptions	445

Revision History

The following table provides the history of changes to this databook.

Date	Version	Description
May 2021	1.51a	<p>Updated:</p> <ul style="list-style-type: none"> ■ “Product Overview” on page 19 ■ “Guidelines for Selecting the Burst or Non-Burst Mode” on page 44 ■ Figure 2-10 on page 53, Figure 2-11 on page 54 ■ Figure 2-12 on page 55, Figure 2-13 on page 56 ■ Figure 2-14 on page 57, Figure 2-15 on page 57 ■ Figure 2-16 on page 59, Figure 2-18 on page 61 ■ Figure 2-19 on page 63, Figure 2-20 on page 66 ■ Figure 2-22 on page 70, Figure 2-59 on page 103 ■ Figure 2-60 on page 104
June 2020	1.50a_sow03	<p>Updated:</p> <ul style="list-style-type: none"> ■ “Parameter Descriptions” on page 113 ■ “Register Descriptions” on page 197 ■ “Signal Descriptions” on page 127 ■ “Internal Parameter Descriptions” on page 459 ■ “Area and Power” on page 417
May 2020	1.50a_ea02	<p>Updated:</p> <ul style="list-style-type: none"> ■ “Parameter Descriptions” on page 113 ■ “Register Descriptions” on page 197 ■ “Signal Descriptions” on page 127 ■ “Area and Power” on page 415 ■ “Internal Parameter Descriptions” on page 457 <p>Added:</p> <ul style="list-style-type: none"> ■ “DWC_mipi_dsi_host Automotive Safety” on page 461

Date	Version	Description
August 2019	1.41a	<p>Updated:</p> <ul style="list-style-type: none"> ■ “Parameter Descriptions” on page 109 ■ “Signal Descriptions” on page 123 ■ “Register Descriptions” on page 193 ■ “Internal Parameter Descriptions” on page 453 ■ “Supported Resolutions and Frame Rates” on page 24 ■ “Speed and Clock Requirements” on page 36 ■ “Transmission of Commands in Low-Power Mode” on page 78 ■ “Errors Raised by Timeouts” on page 103 ■ “Area and Power” on page 411 <p>Added:</p> <ul style="list-style-type: none"> ■ “DSC Rate Buffer RAM Requirements” on page 94 ■ Appendix F, “DWC_mipi_dsi_host Automotive Safety”
April 2018	1.40a	<p>Added:</p> <ul style="list-style-type: none"> ■ “DSC 24 - External DSC Encoder” on page 90 ■ “DSC Encoder - Internal Compression” on page 91 ■ “TE request by Hardware” on page 52 ■ “Auto ULPS control scheme” on page 58 ■ FIFO Status Registers ■ New Appendix D, “Static and Dynamic Registers” on page 475 ■ Combo PHY related updates. ■ Table 1-3 ■ Table 1-4 <p>Updated:</p> <ul style="list-style-type: none"> ■ Figure 1-1 ■ Figure 1-2 ■ Table 1-5 ■ “DWC_mipi_dsi_host Features” on page 19 ■ “Standards Compliance” on page 19 ■ “Supported Resolutions and Frame Rates” on page 21 ■ “Speed and Clock Requirements” on page 31 ■ “Enabling DPI Interface” on page 39 ■ “Updating the DPI Configuration Dynamically” on page 44 ■ “Description of eDPI Interface” on page 48 ■ “Support for Tearing Effect (TE)” on page 51 ■ “DSI Packet Types” on page 61 ■ “Video Mode Pattern Generator Resolution” on page 85 ■ “Display Stream Compression” on page 87 ■ “DWC_mipi_dsi_host Power and Gate Count for Industry Standard 28nm Library” on page 449 <p>The Parameter Descriptions, Signals, and Software Registers chapters are auto-extracted from the RTL.</p>

Date	Version	Description
May 2015	1.31a	<p>Added:</p> <p>“Support for Display Stream Compression” on page 89</p> <p>Updated:</p> <ul style="list-style-type: none"> ■ “Standards Compliance” on page 19 ■ “Supported Resolutions and Frame Rates” on page 21 ■ “Unsupported Features and Exceptions” on page 29 ■ “DWC_mipi_dsi_host Features” on page 19 ■ “Operational Model Overview” on page 29 ■ “Speed and Clock Requirements” on page 31 ■ “Area and Power” on page 32 ■ “DPI Interface” on page 50 ■ “Clock Lane in Low-Power Mode (D-PHY)” on page 82 <p>The Parameter Descriptions, Signals, and Software Registers chapters have been auto-extracted from the RTL.</p>
October 2013	1.30a	<p>Added:</p> <ul style="list-style-type: none"> ■ The section “Video Mode Pattern Generator (D-PHY Only)” on page 85 ■ The section “Updating the DPI Configuration” on page 56 ■ The signal dpiupdatecfg ■ The following registers: <ul style="list-style-type: none"> □ INT_FORCE0 □ INT_FORCE1 □ VID_SHADOW_CTRL □ DPI_VCID_ACT □ DPI_COLOR_CODING_ACT □ DPI_LP_CMD_TIM_ACT □ VID_MODE_CFG_ACT □ VID_PKT_SIZE_ACT □ VID_NUM_CHUNKS_ACT □ VID_NULL_SIZE_ACT □ VID_HSA_TIME_ACT □ VID_HBP_TIME_ACT □ VID_HLINE_TIME_ACT □ VID_VSA_LINES_ACT □ VID_VBP_LINES_ACT □ VID_VFP_LINES_ACT □ VID_VACTIVE_LINES_ACT □ SDF_3D_ACT” on page 197 <p>Continued on next page...</p>

Date	Version	Description
		<p>Updated:</p> <ul style="list-style-type: none"> ■ The register VID_MODE_CFG for video mode pattern generator ■ The following sections: <ul style="list-style-type: none"> □ “Operational Model Overview” on page 29 and Figure 1-2 □ “DWC_mipi_dsi_host Features” on page 19 □ “Error Control” on page 98 □ “DSI Host Generic Signals” on page 115 □ “Video Mode with Stereoscopic Image Data” on page 54 □ “Transmission of Commands in Low-Power Mode” on page 77 ■ Formulas in the “Calculating the Time to Transmit the Commands in LP Mode in the HFP Region” on page 78 ■ The “Value after Reset” of the VERSION, PHY_STATUS, INT_MSK0, and INT_MSK1 registers ■ Table 6-71 Static and Dynamic Registers ■ Figure 2-4 ■ the “Synchronous to” of BISTON and BISTOK signals to Asynchronous. ■ The configuration option “DBI Read FIFO Depth (uses 2-port RAM)” on page 35 ■ The “Registered” item of the interrupt signal description to Yes. <p>Removed the DSI_N_INTERRUPT_PINS configuration option, and interrupt0 and interrupt1 pins.</p>
February 2013	1.21a	<p>Added:</p> <ul style="list-style-type: none"> ■ The section “Reset Structure” ■ Color modes supported by the eDPI in section “eDPI Interface” on page 59 <p>Updated:</p> <ul style="list-style-type: none"> ■ Features list in “DWC_mipi_dsi_host Features” on page 19 ■ Formula in section “Read Command Transmission” on page 81 ■ Corrected the width of the *addr signals and the dependency of the DBI signals in the signal tables of the Signals chapter

Date	Version	Description
August 2012	1.20a	<p>Added:</p> <ul style="list-style-type: none"> ■ A column for the power consumption numbers in Table 1-6. ■ The “Peripheral Response Timeout” section that describes the Peripheral Response Timeout feature (section 7.3.3) of the DSI 1.1 Spec. ■ The descriptions for the new video mode data types of the DSI 1.1v specification in the section “DPI Interface”. ■ The “Video Mode with Stereoscopic Image Data” section that describes the 3D formats. ■ The “Guidelines for Selecting the Burst or Non-Burst Mode” section. ■ The “Clock Lane in Low-Power Mode (D-PHY)” section that describes the support provided to non-continuous clock. ■ The Error Handling Appendix to explain the conditions that trigger the interrupts and provides suggestions to recover from these error states. <p>Updated:</p> <ul style="list-style-type: none"> ■ The “Transmission of Commands in Low-Power Mode” section. ■ The Table 1-5. ■ The size of the dpipixdata[29:0] signal. ■ The descriptions of the presetn, edpte, genericpldwclk, dpipixelrclk, and dbipldrclk signals. ■ The complete Register Descriptions chapter because new configuration registers are added and the dimensions of the existing registers are increased. The register bank is reorganized to meet the requirements of the new features. <p>The reorganization of the register bank impacts the register configuration information in other chapters such as Architecture and Video Interfaces.</p>

Date	Version	Description
April 2012	1.10a	<p>Added:</p> <ul style="list-style-type: none"> ■ Chapter 2, “Building and Verifying DWC_mipi_dsi_host”, which provides an overview of the step-by-step process you use to configure, synthesize, simulate, and export DWC_mipi_dsi_host using the Synopsys coreConsultant tool. ■ Chapter 8, “Integrating DWC_mipi_dsi_host with D-PHY”, which provides an overview of the step-by-step process to integrate the DWC_mipi_dsi_host with your D-PHY and the details for FPGA prototyping of the DWC_mipi_dsi_host. ■ Appendix B, “DBI Color Code Mapping Waveforms”. ■ The FIFO depth information to the parameters in Table 2-2. ■ The new parameter DSI_HOST_FPGA in “Configuration Parameters” chapter. ■ The “eDPI Interface”, “Transmission of Commands”, “Selection of Appropriate Interface”, and “Virtual Channels” sections in the “Video Interfaces” chapter. ■ The “EDPI_CFG” and “LP_CMD_TIM” registers in the “Register Descriptions” chapter. ■ The dpivsync_edpiwms, edpihalt, and edpite pins. ■ The new testcases test_vtb_dpi_generic_lpcmd and test_vtb_edpi_01 in Table 9-2. ■ The eDPI task “write_edpi_packet”. ■ Table 3-2 that shows number of pixels to bytes conversion for the DBI pixels formats. <p>Updated:</p> <ul style="list-style-type: none"> ■ The “Speed and Clock Requirements” section of the “Product Overview” chapter. ■ The “DWC_mipi_dsi_host Features” section of the “Product Overview” chapter. ■ The names of environment variables for Synopsys PHY. ■ The supported display settings in Table 1-1. ■ The area numbers in Table 1-6. ■ The “GEN_HDR”, “CLKMGR_CFG?”, “PHY_TMR_CFG”, and “GEN_PLD_DATA” registers in the “Register Descriptions” chapter. ■ The DSI_DATAINTERFACE parameter in Table 2-2.
July 2011	1.01a	<p>Added:</p> <ul style="list-style-type: none"> ■ A new signal phyulpsactivenotclk. ■ The description of the flow to instruct D-PHY to enter and exit ULPM. ■ The “DWC_mipi_dsi_host Verification Environment” chapter. ■ The section “Partitioning the Long write_memory_start Commands” in the “Video Interfaces” chapter. ■ Table 1-1, which shows some of the supported display settings. <p>Removed:</p> <ul style="list-style-type: none"> ■ All references to Asynchronous RAMs as DSI controller now supports both Synchronous and Asynchronous RAMs. <p>Updated:</p> <ul style="list-style-type: none"> ■ The value after reset of the register Version to 0x3130312A. ■ The area values in Table 1-5.
December 2010	1.00a	Initial release

Preface

This databook describes the DesignWare Cores MIPI DSI Host Controller (DWC_mipi_dsi_host controller), which along with Synopsys DWC MIPI D-PHY or combo-PHY is a part of the complete MIPI DSI solution.

Databook Organization

The chapters of this databook are organized as follows:

- [Chapter 1, "Product Overview"](#), provides an introduction to DWC_mipi_dsi_host controller, including a block diagram, supported standards and features, clock and memory requirements, and so on.
- [Chapter 2, "Architecture"](#), describes the general architecture, interfaces, transmission of commands, virtual channels, and macro structure of DWC_mipi_dsi_host controller.
- [Chapter 3, "Parameter Descriptions"](#), describes the hardware configuration parameters.
- [Chapter 4, "Signal Descriptions"](#), provides descriptions of the inputs or outputs of the controller.
- [Chapter 5, "Register Descriptions"](#), provides the memory map of DWC_mipi_dsi_host controller and the descriptions of the programmable software registers.
- [Appendix A, "Area and Power"](#), provides the area and power numbers of the DWC_mipi_dsi_host controller for different configurations.
- [Appendix B, "DBI Color Code Mapping Waveforms"](#), contains the diagrams that depict how the pixel-to-byte conversion is done for each color code mapping supported by the Display Bus Interface (DBI).
- [Appendix C, "Error Handling"](#), explains the conditions that trigger the interrupts and provides suggestions to recover from these error states.
- [Appendix D, "Static and Dynamic Registers"](#), lists the dynamic and static registers of the controller.
- [Appendix E, "Internal Parameter Descriptions"](#), provides a description of the internal parameters that might be indirectly referenced in expressions in the Signals, Parameters, or Registers chapters.

Related Documentation

Refer to the following documentation:

- coreConsultant User's Guide
- coreAssembler User's Guide

Web Resources

- DesignWare IP product information: <https://www.synopsys.com/designware-ip.html>
- Your custom DesignWare IP page: <https://www.synopsys.com/dw/mydesignware.php>
- Documentation through SolvNet: <https://solvnetplus.synopsys.com> (Synopsys password required)
- Synopsys Common Licensing (SCL): <http://www.synopsys.com/keys>

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

Customer Support

To obtain support for your product, prepare the required files and contact the support center using one of the methods described:

- Prepare the following debug information, if applicable:
 - For environment set-up problems or failures with configuration, simulation, or synthesis that occur within coreConsultant or coreAssembler, select the following menu:
 - **File > Build Debug Tar-file**
 - For simulation issues outside of coreConsultant or coreAssembler:
 - Create a waveform file (such as VPD or VCD).
 - Identify the hierarchy path to the DesignWare instance.
 - Identify the timestamp of any signals or locations in the waveforms that are not understood.
 - For the fastest response, enter a case through SolvNetPlus:
 - a. <https://solvnetplus.synopsys.com>



SolvNetPlus does not support Internet Explorer. Use a supported browser such as Microsoft Edge, Google Chrome, Mozilla Firefox, or Apple Safari.

- b. Click the **Cases** menu and then click **Create a New Case** (below the list of cases).
- c. Complete the mandatory fields that are marked with an asterisk and click **Save**.

Make sure to include the following:

- **Product L1:** DesignWare Cores

- **Product L2:** MIPI Controller

- d. After creating the case, attach any debug files you created.

For more information about general usage information, refer to the following article in SolvNet-Plus:

<https://solvnetplus.synopsys.com/s/article/SolvNetPlus-Usage-Help-Resources>

- Or, send an e-mail message to support_center@synopsys.com (your email will be queued and then, on a first-come, first-served basis, manually routed to the correct support engineer):

- ❑ Include the Product L1 and Product L2 names, and Version number in your e-mail so it can be routed correctly.
 - ❑ For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood
 - ❑ Attach any debug files you created.

- Or, telephone your local support center:

- ❑ North America:

Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.

- ❑ All other countries:

<https://www.synopsys.com/support/global-support-centers.html>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

Product Overview

This chapter contains the following sections:

- “General Product Description” on page 20
- “Related Synopsys Products” on page 35

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

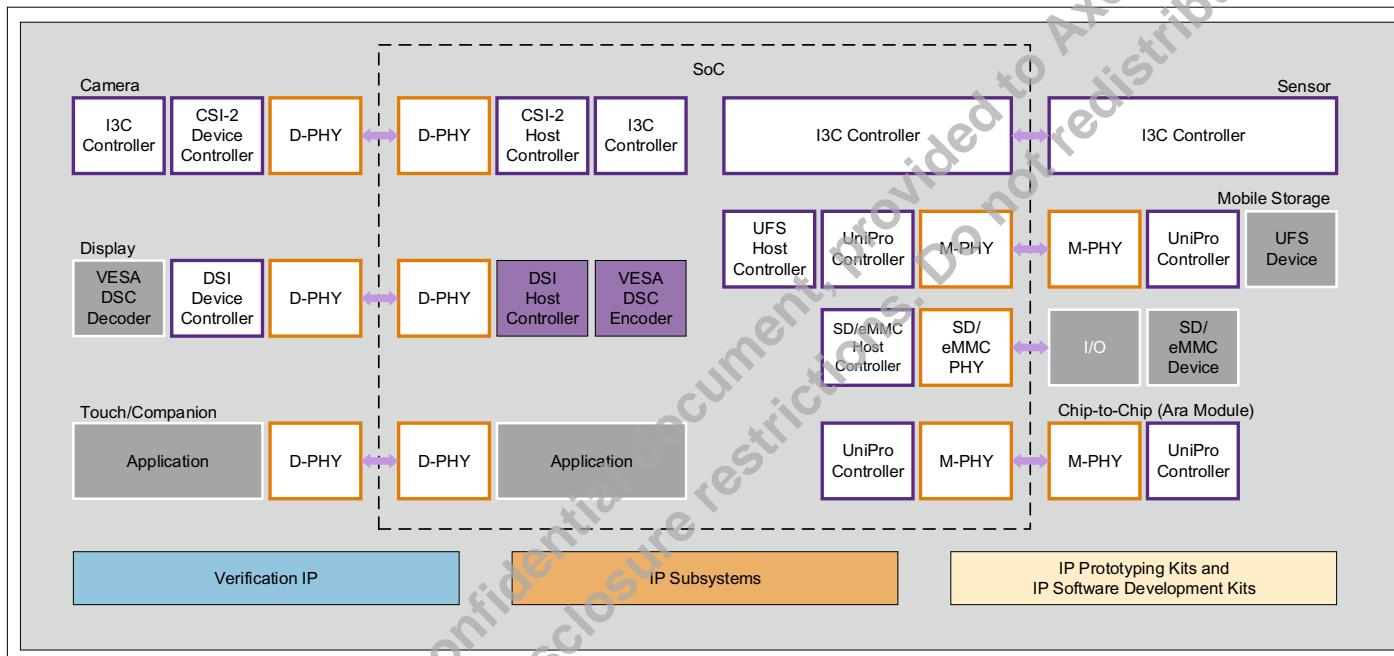
1.1 General Product Description

The Display Serial Interface (DSI) is part of a group of communication protocols defined by the MIPI Alliance. The DesignWare Cores MIPI DSI Host Controller (referred to as `DWC_mipi_dsi_host_controller`) is a digital controller that implements all protocol functions defined in the MIPI DSI Specification.

The `DWC_mipi_dsi_host_controller` provides an interface between the system and the MIPI D-PHY or Combo PHY (D-PHY and C-PHY), allowing the communication with a DSI-compliant display. There is a broad range of D-PHY IPs that includes bidirectional PHYs with two and four lanes for several technologies.

For more information about MIPI D-PHY IP, visit the [Synopsys DesignWare MIPI D-PHY IP Solution](#) page. [Figure 1-1](#) shows the `DWC_mipi_dsi_host_controller` in an example system-on-chip design.

Figure 1-1 DWC_mipi_dsi_host Controller in System-on-Chip Example



1.1.1 Standards Compliance

The `DWC_mipi_dsi_host_controller` conforms to the following standards:

- MIPI® Alliance Specification for Display Serial Interface (DSI) Version 1.2 - 16 June 2014
- MIPI® Alliance Standard for Display Bus Interface v2.00 (DBI-2) - 16 Nov 2005
- MIPI® Alliance Specification for Display Command Set (DCS) Version 1.3 - 5 October 2015
- MIPI® Alliance Standard for Display Pixel Interface v2.00 (DPI-2) - 15 September 2005
- MIPI® Alliance Specification for Stereoscopic Display Formats (SDF) v1.0 - 22 November 2011
- MIPI® Alliance Specification for D-PHY v1.2, 01 August 2014
- MIPI® Alliance Specification for C-PHY v1.0, October 2014
- AMBA 2.0 Specification (APB) from ARM
- VESA Display Stream Compression (DSC) Standard Version 1.1 - 1 August 2014

1.1.2 Supported Features

MIPI DSI Related Features

The DWC_mipi_dsi_host controller supports the following features from DSI Specification:

- Compliant with MIPI Alliance standards (For more information, see “[Standards Compliance](#)” on page [20](#))
- Interface with MIPI D-PHY through PHY Protocol Interface (PPI), as defined in MIPI Alliance specification for D-PHY
- Interface with MIPI C-PHY through PHY Protocol Interface (PPI), as defined in MIPI Alliance Specification for C-PHY. Only Combo PHY mode supports this feature. Support for Combo PHY mode requires an additional license.
- All commands defined in the MIPI Alliance specification for Display Command Set (DCS):
 - Transmission of all Command mode packets through an Advanced Peripheral Bus (APB) interface
 - Transmission of commands in low-power and high-speed, during Video mode
- When in D-PHY, supports:
 - Up to 2.5 Gsps per lane
 - Up to four data lanes
 - Non-continuous and Continuous modes
- When in C-PHY, supports:
 - Up to 2.5 Gsps per lane
 - Continuous (expected in single-lane mode) and Non-continuous modes
- When in Combo PHY, supports:
 - Four data lanes on D-PHY, and three data lanes on C-PHY
- Bidirectional communication and escape mode support through data lane 0
- ECC and checksum capabilities
- End of Transmission Packet (EoTp)
- 3D Control Transmission (For more information on Stereoscopic Display Control in Video mode, see DSI Specification)
- DBI features:
 - Configurable DBI types:
 - Type A interface including Tearing Effect and Fixed E mode
 - Type A interface including Tearing Effect and Clocked E mode
 - Type B interface including Tearing Effect
 - Extended pixel clock speed beyond the DBI standard maximum clock of 20 MHz. The value can go up to D-PHY clock lane speed divided by 12. For example, for a maximum D-PHY clock lane speed of 500 MHz, the maximum DBI clock is 41.7 MHz
 - DBI color coding mappings:
 - 8-bit interface: 8, 12, 16, 18, and 24 bits per pixel

- 9-bit interface: 18 bits per pixel
- 16-bit interface: 8, 12, 16, 18 (options 1 and 2), and 24 (options 1 and 2) bits per pixel
- DPI features:
 - DPI color coding mappings into 30-bit interface:
 - 16-bit RGB, configurations 1, 2, and 3
 - 18-bit RGB, configurations 1 and 2
 - 24-bit RGB
 - 30-bit RGB
 - 36-bit RGB (double clock rate required)
 - 24-bit YCbCr 4:2:2
 - 20-bit YCbCr 4:2:2 loosely packed
 - 16-bit YCbCr 4:2:2
 - 12-bit YCbCr 4:2:0
 - DSC24 compressed data
- VESA Display Stream Compression (DSC)
 - Supports Internal & External DSC encoder
 - DSC configuration and VESA PPS through register bank
 - Support for compression over DPI for Video and Command mode displays with RGB 24-bit / 30-bit
 - RGB 24-bit (8bpc): 8bpp (DSI v1.2 Spec Annex D - Profile 8: 3X compression) and 12bpp (DSI v1.2 Spec Annex D - Profile 12: 2X compression)
 - RGB 30-bit (10bpc): 10bpp (DSI v1.2 Spec Annex D - Generic Profile: 3X compression) and 12bpp (DSI v1.2 Spec Annex D - Generic Profile: 2.5X compression)
 - Supports horizontal and vertical slices within a frame

Synopsys Specific Features

The DWC_mipi_dsi_host controller supports the following Synopsys specific features:

- Configurable selection of system interfaces:
 - AMBA APB for control and optional support for Generic and DCS commands
 - Display Pixel Interface (DPI) for Video mode interface (optional)
 - Display Bus Interface (DBI) for Command mode interface (optional)
 - Enhanced Display Pixel Interface (eDPI) for Video and Command mode interface (optional)
 - Independently programmable virtual channel ID in DBI, DPI, eDPI, and APB Slave (optional)
 - Display Stream Compression (DSC) encoder (requires additional license)
- Fault recovery mechanisms
- Ultra Low-Power mode with PLL disabled
- DPI features:

- ❑ DPI signals programmable polarity
- ❑ Extended resolutions beyond the DPI standard maximum resolution of 800x480 pixels
- ❑ Maximum resolution limited by available DSI physical link bandwidth, which is determined by the number of physical lanes and the maximum speed that they can achieve
 - For a list of supported example resolutions, see "[Resolution and Frame Rates](#)" on page [26](#)
- Co-existence of DBI and DPI, contingent upon their operability being non-simultaneous
- Enhanced DPI (eDPI) features:
 - ❑ Supports all DPI features (Standard Video mode)
 - ❑ Additionally, enables transferring large amounts of data through the memory_write_start (WMS) and memory_write_continue (WMC) DCS commands (Adapted Command mode)
 - ❑ eDPI (Adapted Command mode) supported color coding formats:
 - 16-bit RGB, configurations 1, 2, and 3
 - 18-bit RGB, configurations 1 and 2
 - 24-bit RGB
- Video pattern generator:
 - ❑ Vertical and horizontal color bar generation without DPI stimuli
 - ❑ BER pattern without DPI stimuli
- Auto ULPS control mechanism
- Tear effect request by hardware

1.1.3 Unsupported Features

MIPI DSI Related Features

The DWC_mipi_dsi_host controller does not support the following features:

- The System Level DBI Type C Serial Interface as defined in MIPI Alliance Standard for Display Bus Interface is not supported in this version of the DWC_mipi_dsi_host controller.
- DSI split links, which enables a horizontal scanline of active pixels to be divided into two or more streams.

1.1.4 Applications

Typical applications built with the DWC_mipi_dsi_host controller are as follows:

- Handheld devices
- Smartphone
- Multimedia tablets
- MID
- Navigation
- Gaming consoles
- DVC

- Automotive
- IoT
- Wearables

1.1.5 Requirements

1.1.5.1 Clock

[Table 1-1](#) lists the validated clock frequencies of the DWC_mipi_dsi_host controller.

Table 1-1 **DWC_mipi_dsi_host Controller Clock Frequencies**

Clock Domain	Minimum Frequency	Maximum Frequency
rxclkescl	Not applicable	<ul style="list-style-type: none"> ■ The maximum frequency is 20 MHz ■ Limited by MIPI D-PHY/C-PHY Specification
txclkescl	Not applicable	<ul style="list-style-type: none"> ■ The maximum frequency is 20 MHz ■ Limited by MIPI D-PHY/C-PHY Specification. If required, this value can be exceeded
lanebyteclk	The minimum frequency should be three times rxclkescl	<p>This frequency is 1/8th of the D-PHY maximum speed. The maximum speed that can be used also depends on the standard cells library that is used for synthesis.</p> <ul style="list-style-type: none"> ■ 1.0 Gbps D-PHY: Maximum frequency of lanebyteclk = 125.0 MHz ■ 1.5 Gbps D-PHY: Maximum frequency of lanebyteclk = 187.5 MHz ■ 2.5 Gbps D-PHY: Maximum frequency of lanebyteclk = 312.5 MHz <p>In C-PHY mode: This frequency is 1/7th of the C-PHY maximum speed. The maximum speed that can be used also depends on the standard cells library that is used for synthesis.</p> <ul style="list-style-type: none"> ■ 1.0 Gps C-PHY: Maximum frequency of lanebyteclk = 142.86 MHz ■ 1.5 Gps C-PHY: Maximum frequency of lanebyteclk = 214.29 MHz ■ 2.5 Gps C-PHY: Maximum frequency of lanebyteclk = 357.14 MHz
pclk	5 MHz	The DWC_mipi_dsi_host controller can support several technologies up to 220 MHz. The maximum speed that can be used depends on the standard cells library that is used for synthesis.
dipiclk	2 MHz	The DWC_mipi_dsi_host controller can support several technologies up to 500 MHz. The maximum speed that can be used depends on the standard cells library that is used for synthesis.

Clock Domain	Minimum Frequency	Maximum Frequency
dipiclk_dsc	Not applicable	<p>The frequency of dpiclk_dsc should be equal to:</p> <ul style="list-style-type: none"> ■ dpiclk / 2 for dual dsc mode ■ dpiclk for single dsc mode <p>It is also necessary to ensure that both the clocks are synchronous.</p>
dbiclk	Not applicable	<ul style="list-style-type: none"> ■ The frequency of dbiclk is restricted to at least three times less than lanebyteclk. The dbiclk signal uses the same asynchronous clock source as the DBI. It corresponds to the following signals: <ul style="list-style-type: none"> □ CSX for type A fixed E interface □ E for type A clocked E interface □ WRX/RDX for type B interface

1.1.5.2 Memory

Depending on the chosen hardware configuration, the DWC_mipi_dsi_host controller may require several RAMs to operate correctly. For more information, see “[Supported 2-Port RAM Types](#)” on page [99](#), and “FIFOs Configuration Parameters” section in “Parameter Descriptions” chapter.

DPI/eDPI Requirements

The memory requirements for DPI/eDPI are as follows:

- **DPI/eDPI Burst Video Mode:** For more information on memory requirements in DPI/eDPI Burst Video Mode, see “[Burst Mode](#)” on page [44](#).
- **DPI/eDPI Non-Burst Video Mode:** For more information on memory requirements in DPI/eDPI Non-Burst Video Mode, see “[Non-Burst Mode](#)” on page [44](#).

DBI Requirements

If automatic partitioning is a system requirement, see “[Dimensioning the Size of FIFOs for Automatic Partitioning](#)” on page [68](#) for more information.

APB Requirements

For more information on APB memory requirements, see “[Packet Transmission Using the Generic Interface](#)” on page [72](#).

DSC Requirements

For more information on DSC memory requirements, see “[DSC Rate Buffer RAM Requirements](#)” on page [94](#).

1.1.6 Limitations and Exceptions

The DWC_mipi_dsi_host controller has the following limitations:

- For the DBI, it is necessary to operate lanebyteclk three times faster than dbiclk.
- VTB (Verification TestBench) with internal VESA DSC encoder only supports DPI Video mode.

- Minimum horizontal porch supported is 80 DPIPCLK cycles when using internal VESA DSC encoder.

1.1.7 Resolution and Frame Rates

The DSI specification does not define supported standard resolutions or frame rates.

The following play a fundamental role in the required bandwidth:

- Display resolution
- Blanking periods
- Synchronization events duration
- Frame rates
- Pixel color depth

Additionally, the following link related attributes influences the ability of the link to support a DSI-specific device:

- Display input buffering capabilities
- Video transmission mode (Burst or Non-Burst)
- Bus Turn-Around time
- Concurrent command mode traffic in a video mode transmission
- Display device specifics

All these variables make it difficult to define a standard procedure to estimate the minimum lane rate and the minimum number of lanes that support a specific display device.

Table 1-2 Supported Video Mode 1

Display Type							DSI Controller					
Resolution (pixels)	Horizontal		Vertical		Refresh Rate (Hz)	Color Depth (bpp)	DPI/eDPI		Number of Lanes D-PHY		Number of trios C-PHY	
	Active	Total	Active	Total			DSI Bandwidth (Mbps)	DPI Clock (MHz)	1Gbps	2.5Gbps	1Gbps	2.5Gbps
320x240	320	380	240	246	60	24	135	5.6	1	1	1	1
640x480	640	800	480	525	60	24	605	25.2	1	1	1	1
800x600	800	1056	600	628	30	24	477	19.9	1	1	1	1
800x600	800	1056	600	628	60	24	955	39.8	1	1	1	1
1024x768	1024	1344	768	806	30	24	780	32.5	1	1	1	1
1024x768	1024	1344	768	806	60	24	1560	65.0	2	1	1	1
1280x800	1280	1688	800	831	60	24	2020	84.2	3	1	1	1
1280x1024	1280	1688	1024	1066	60	24	2591	108.0	3	2	2	1
1600x900	1600	1800	900	1000	60	24	2592	108.0	3	2	2	1
1680x1050	1680	1840	1050	1080	60	24	2862	119.2	3	2	2	1
1920x1080	1920	2576	1080	1120	30	24	2077	86.6	3	1	1	1
1920x1080 RB	1920	2080	1080	1111	60	24	3328	138.7	4	2	2	1
1920x1080	1920	2576	1080	1120	60	24	4155	173.1	N/S	2	2	1
1920x1200	1920	2592	1200	1245	30	24	2323	96.8	3	1	2	1
1920x1200	1920	2592	1200	1245	60	24	4647	193.6	N/S	2	3	1
1920x1200 RB	1920	2080	1200	1235	60	24	3699	154.1	4	2	2	1
2048X1080 (2K)	2048	2200	1080	1125	60	24	3564	148.5	4	2	2	1

Display Type							DSI Controller					
Resolution (pixels)	Horizontal		Vertical		Refresh Rate (Hz)	Color Depth (bpp)	DPI/eDPI		Number of Lanes D-PHY		Number of trios C-PHY	
	Active	Total	Active	Total			DSI Bandwidth (Mbps)	DPI Clock (MHz)	1Gbps	2.5Gbps	1Gsps	2.5Gsps
3840x2160 RB	3840	4000	2160	2191	30	24	6310	262.9	N/S	3	3	2
3840x2160 RB	3840	4000	2160	2191	60	24	12620	525.8	N/S	N/S	N/S	3
3840x2160 (UHD)	3840	4400	2160	2250	60	24	14256	594.0	N/S	N/S	N/S	3
4096X2160 (4K)	4096	4400	2160	2250	60	24	14256	594.0	N/S	N/S	N/S	3

Table 1-3 Supported Video Mode With DSC

Display Type							DSI Controller with internal DSC ¹						
Resolution (pixels)	Horizontal		Vertical		Refresh Rate (Hz)	Color Depth (bpp)	DPI/eDPI			Number of Lanes D-PHY		Number of trios C-PHY	
	Active	Total	Active	Total			DSI Bandwidth (Mbps) ²	DPI Clock (MHz) Dual DSC Dual Port	DPI Clock (MHz) Other DSC Options	1Gbps	2.5Gbps	1Gsp	2.5Gsp
320x240	320	380	240	246	60	24	45	2.8	5.6	1	1	1	1
640x480	640	800	480	525	60	24	202	12.6	25.2	1	1	1	1
800x600	800	1056	600	628	30	24	159	9.9	19.9	1	1	1	1
800x600	800	1056	600	628	60	24	318	19.9	39.8	1	1	1	1
1024x768	1024	1344	768	806	30	24	260	16.2	32.5	1	1	1	1
1024x768	1024	1344	768	806	60	24	520	32.5	65.0	1	1	1	1
1280x800	1280	1688	800	831	60	24	673	42.1	84.2	1	1	1	1
1280x1024	1280	1688	1024	1066	60	24	864	54.0	108.0	1	1	1	1
1600x900	1600	1800	900	1000	60	24	864	54.0	108.0	1	1	1	1
1680x1050	1680	1840	1050	1080	60	24	954	59.6	119.2	1	1	1	1
1920x1080	1920	2576	1080	1120	30	24	692	43.3	86.6	1	1	1	1
1920x1080 RB ³	1920	2080	1080	1111	60	24	1109	69.3	138.7	2	1	1	1
1920x1080	1920	2576	1080	1120	60	24	1385	86.6	173.1	2	1	1	1
1920x1200	1920	2592	1200	1245	30	24	774	48.4	96.8	1	1	1	1
1920x1200	1920	2592	1200	1245	60	24	1549	96.8	193.6	2	1	1	1

Display Type							DSI Controller with internal DSC ¹						
Resolution (pixels)	Horizontal		Vertical		Refresh Rate (Hz)	Color Depth (bpp)	DPI/eDPI			Number of Lanes D-PHY		Number of trios C-PHY	
	Active	Total	Active	Total			DSI Bandwidth (Mbps) ²	DPI Clock (MHz) Dual DSC Dual Port	DPI Clock (MHz) Other DSC Options	1Gbps	2.5Gbps	1Gsp	2.5Gsp
1920x1200 RB	1920	2080	1200	1235	60	24	1233	77.1	154.1	2	1	1	1
2048X1080 (2K)	2048	2200	1080	1125	60	24	1188	74.3	148.5	2	1	1	1
3840x2160 RB	3840	4000	2160	2191	30	24	2103	131.5	262.9	3	1	1	1
3840x2160 RB	3840	4000	2160	2191	60	24	4207	262.9	525.8	N/S	2	2	1
3840x2160 (UHD)	3840	4400	2160	2250	60	24	4752	297.0	594.0	N/S	2	3	1
4096X2160 (4K)	4096	4400	2160	2250	60	24	4752	297.0	594.0	N/S	2	3	1

1. Compression factor 3 - 1

2. Blanking also reduced in same proportion of the compression

3. RB - Reduced Blanking

Table 1-4 Supported Command Mode 1

Display Type					DSI Controller with eDPI						Raw DBI Clock (MHz) D-PHY *		Raw APB Slave Clock (MHz)	
Resolution (pixels)	Horizontal	Vertical	Refresh Rate (Hz)	Color Depth (bpp)	eDPI		Number of Lanes D-PHY		Number of trios C-PHY					Raw APB Slave Clock (MHz)
					DSI Bandwidth (Mbps) ¹	DPI Clock (MHz) ²	1Gbps	2.5Gbps	1Gsp	2.5Gsp	1Gbps	2.5Gbps	1Gbps	
320x240	320	240	60	24	111	4.6	1	1	1	1	6.9	6.9	7	
640x480	640	480	60	24	442	18.4	1	1	1	1	27.6	27.6	28	
800x600	800	600	30	24	346	14.4	1	1	1	1	21.6	21.6	22	
800x600	800	600	60	24	691	28.8	1	1	1	1	N/S	43.2	43	
1024x768	1024	768	30	24	566	23.6	1	1	1	1	35.4	35.4	35	
1024x768	1024	768	60	24	1132	47.2	2	1	1	1	N/S	70.8	71	
1280x800	1280	800	60	24	1475	61.4	2	1	1	1	N/S	92.2	92	
1280x1024	1280	1024	60	24	1887	78.6	2	1	1	1	N/S	N/S	118	
1600x900	1600	900	60	24	2074	86.4	3	1	1	1	N/S	N/S	130	
1680x1050	1680	1050	60	24	2540	105.8	3	2	2	1	N/S	N/S	159	
1920x1080	1920	1080	30	24	1493	62.2	2	1	1	1	N/S	93.3	93	
1920x1080 RB	1920	1080	60	24	2986	124.4	3	2	2	1	N/S	N/S	187	
1920x1080	1920	1080	60	24	2986	124.4	3	2	2	1	N/S	N/S	187	
1920x1200	1920	1200	30	24	1659	69.1	2	1	1	1	N/S	103.7	104	
1920x1200	1920	1200	60	24	3318	138.2	4	2	2	1	N/S	N/S	207	
1920x1200 RB	1920	1200	60	24	3318	138.2	4	2	2	1	N/S	N/S	207	

Display Type					DSI Controller with eDPI							Raw DBI Clock (MHz) D-PHY *		Raw APB Slave Clock (MHz)	
Resolution (pixels)	Horizontal	Vertical	Refresh Rate (Hz)	Color Depth (bpp)	eDPI		Number of Lanes D-PHY		Number of trios C-PHY						
					DSI Bandwidth (Mbps) ¹	DPI Clock (MHz) ²	1Gbps	2.5Gbps	1Gsps	2.5Gsps	1Gbps	2.5Gbps			
2048X1080 (2K)	2048	1080	60	24	3185	132.7	4	2	2	1	N/S	N/S	199		
3840x2160 RB ³	3840	2160	30	24	5972	248.8	N/S	3	3	2	N/S	N/S	373		
3840x2160 RB	3840	2160	60	24	11944	497.7	N/S	N/S	N/S	3	N/S	N/S	746		
3840x2160 (UHD)	3840	2160	60	24	11944	497.7	N/S	N/S	N/S	3	N/S	N/S	746		
4096X2160 (4K)	4096	2160	60	24	12740	530.8	N/S	N/S	N/S	3	N/S	N/S	796		

1. Minimum value will be considered only active period. Low-power to high-speed and high-speed to low-power periods need to be added. If recommended blanking is used, please check Video Mode

2. The number of Lanes for DBI is the same as eDPI. DBI is only supported with D-PHY

3. RB - Reduced Blanking

Table 1-5 Supported Command Mode 2

Display Type					DSI Controller with internal DSC						
Resolution (pixels)	Horizontal	Vertical	Refresh Rate (Hz)	Color Depth (bpp)	DPI/eDPI			Number of Lanes D-PHY		Number of trios C-PHY	
					DSI Bandwidth (Mbps) ¹	DPI Clock (MHz) Dual DSC Dual Port ²	DPI Clock (MHz) Other DSC Options	1Gbps	2.5Gbps	1Gsps	2.5Gsps
320x240	320	240	60	24	37	2.3	4.6	1	1	1	1
640x480	640	480	60	24	147	9.2	18.4	1	1	1	1
800x600	800	600	30	24	115	7.2	14.4	1	1	1	1
800x600	800	600	60	24	230	14.4	28.8	1	1	1	1
1024x768	1024	768	30	24	189	11.8	23.6	1	1	1	1
1024x768	1024	768	60	24	377	23.6	47.2	1	1	1	1
1280x800	1280	800	60	24	492	30.7	61.4	1	1	1	1
1280x1024	1280	1024	60	24	629	39.3	78.6	1	1	1	1
1600x900	1600	900	60	24	691	43.2	86.4	1	1	1	1
1680x1050	1680	1050	60	24	847	52.9	105.8	1	1	1	1
1920x1080	1920	1080	30	24	498	31.1	62.2	1	1	1	1
1920x1080 RB	1920	1080	60	24	995	62.2	124.4	1	1	1	1
1920x1080	1920	1080	60	24	995	62.2	124.4	1	1	1	1
1920x1200	1920	1200	30	24	553	34.6	69.1	1	1	1	1
1920x1200	1920	1200	60	24	1106	69.1	138.2	2	1	1	1
1920x1200 RB	1920	1200	60	24	1106	69.1	138.2	2	1	1	1

Display Type					DSI Controller with internal DSC						
Resolution (pixels)	Horizontal	Vertical	Refresh Rate (Hz)	Color Depth (bpp)	DPI/eDPi			Number of Lanes D-PHY		Number of trios C-PHY	
					DSI Bandwidth (Mbps) ¹	DPI Clock (MHz) Dual DSC ₂	DPI Clock (MHz) Other DSC Options	1Gbps	2.5Gbps	1Gbps	2.5Gbps
2048X1080 (2K)	2048	1080	60	24	1062	66.4	132.7	2	1	1	1
3840x2160 RB ³	3840	2160	30	24	1991	124.4	248.8	2	1	1	1
3840x2160 RB	3840	2160	60	24	3981	248.8	497.7	4	2	2	1
3840x2160 (UHD)	3840	2160	60	24	3981	248.8	497.7	4	2	2	1
4096X2160 (4K)	4096	2160	60	24	4247	265.4	530.8	N/S	2	2	1

1. Minimum value will be considered only active period. Low-power to high-speed and high-speed to low-power periods need to be added. If recommended blanking is used, please check Video Mode

2. The number of Lanes for DBI is the same as eDPi. DBI is only supported with D-PHY

3. RB - Reduced Blanking

1.2 Related Synopsys Products

The various Synopsys products such as the Synopsys DesignWare MIPI DSI Host and Device controllers, C-PHY/D-PHY, and D-PHY IP provides a complete display interface IP solution. This helps in lowering the risk and cost of integration of the MIPI DSI interfaces into application processors, display bridge integrated circuits (ICs), and multimedia co-processors, while improving the time-to-market.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

Architecture

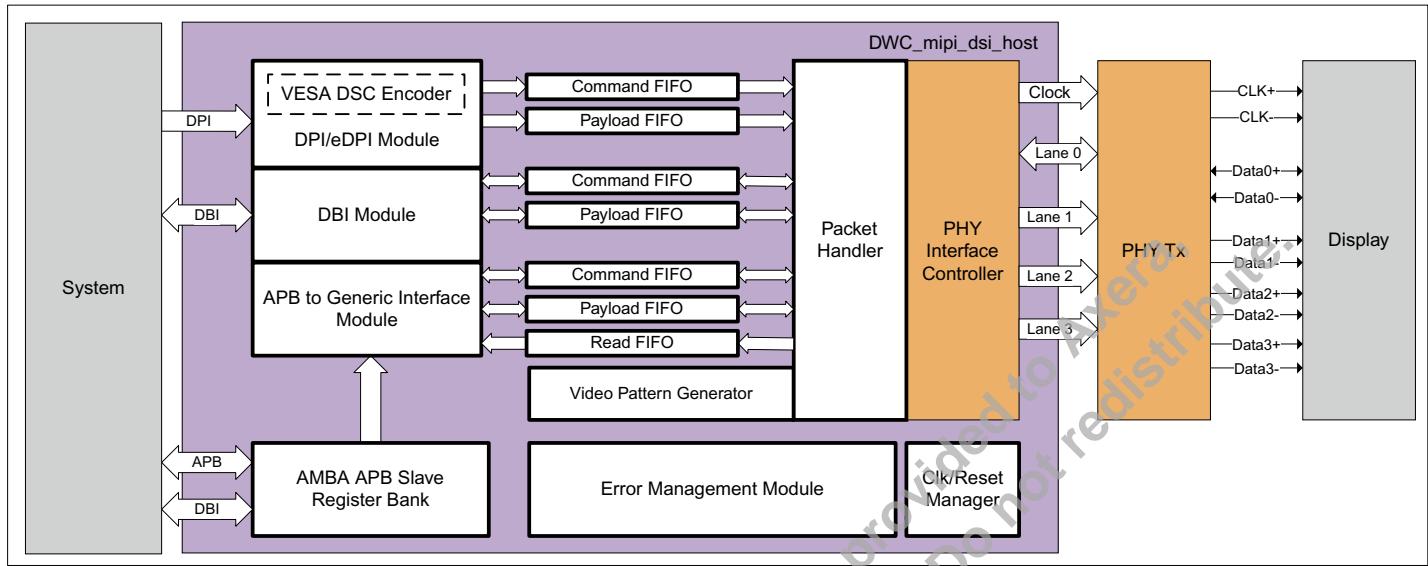
This chapter describes the general architecture and the different parts of the DWC_mipi_dsi_host controller, the supported RAM types, timers, and the error control. It has the following sections:

- “Overview of Architecture” on page 38
- “Selection of Appropriate Interface” on page 40
- “DPI” on page 41
- “eDPI” on page 50
- “DBI” on page 64
- “Continuous Clock Using C-PHY Mode” on page 70
- “APB Slave Generic Interface” on page 71
- “Transmission of Commands” on page 76
- “Virtual Channels” on page 85
- “Video Mode Pattern Generator” on page 87
- “Display Stream Compression” on page 92
- “Supported 2-Port RAM Types” on page 99
- “Timeout Timers” on page 100
- “Error Control” on page 105

2.1 Overview of Architecture

Figure 2-1 shows the architecture of the DWC_mipi_dsi_host controller.

Figure 2-1 DWC_mipi_dsi_host Controller Architecture



The main blocks of the DWC_mipi_dsi_host controller are as follows:

- **DPI/eDPI Module:** Captures the data and control signals and conveys them to two dedicated FIFOs, one for the video control signals, and the other for the pixel data. This data is used to build one of the following:
 - Video packets, in Video mode (For more information, see “[DPI](#)” on page [41](#))
 - DSC commands (memory_write_start (WMS), and memory_write_continue (WMC)), in Command mode (For more information, see “[eDPI](#)” on page [50](#))
- **VESA DSC Encoder:** The VESA DSC encoder implements the VESA Display Stream Compression algorithm. Setting the DSI_HOST_DSC_ENC parameter to 1, enables the VESA DSC encoder. When enabled, it compresses the image data internally, and enables support for RGB24 and RGB30 formats. The VESA DSC encoder delivers the compressed data in either Video mode or in Command mode. When using VESA DSC Encoder, ensure to program all the required registers, such as the DSC_MODE, DSC_FIFO_*, DSC_ENC_*, and DSC_ENC_PPS_* registers.
- **DBI Module:** Encapsulates DCS commands in DSI packets, and conveys these packets to the command and payload FIFOs. For commands that require a response from the device, the controller architecture has a dedicated data FIFO. For more information about DBI, see “[DBI](#)” on page [64](#).
- **Register Bank:** Use the standard AMBA-APB slave interface to access the Register Bank. Program the DWC_mipi_dsi_host registers through the AMBA-APB slave interface to configure and control the behaviour of the DWC_mipi_dsi_host controller. The DWC_mipi_dsi_host controller provides a fully programmable interrupt generator to manage certain specific events.
- **PHY Interface Controller:** Manages PPI, by acknowledging the current operation and enables the low-power transmission or reception, or high-speed transmission. It also performs data splitting between available PHY lanes for high-speed transmission.
- **Packet Handler:** Based on the controller interface configuration (DBI, DPI, or eDPI), and Video Transmission mode (burst or non-burst, with sync pulses or sync events), this block generates packets with ECC and CRC codes. This block also performs the following functions:

- ❑ Packet reception
- ❑ Packet header validation by checking the ECC
- ❑ Packet header correction and notification for single-bit errors
- ❑ Determine termination of reception
- ❑ Error management (detection and notification of errors, for example, multiple header error)
- ❑ Packet routing mechanism, for example, this block analyzes the packet's virtual channel, and routes the output data to the respective port (Generic or DBI).
- **APB-to-Generic Interface Module:** Bridges APB operations and FIFOs holding the Generic commands, that is, the command FIFO, write-payload FIFO, and the read-payload FIFO.
- **Error Management Module:** Monitors the system error conditions, and triggers an interrupt to prompt an error notification.

2.2 Selection of Appropriate Interface

Consider the following factors while selecting the appropriate interface for your system.

DBI

Displays that support Command mode for DCS commands use this interface. The Command mode display device contains its own video timing control generation and a frame buffer. The controller fills the frame buffer of the device with pixel write commands, when the device displays the content of the frame buffer on the display. The device also contains several commands as defined in the DCS Specification, to control its operation. These commands can be either read or write commands, and therefore, DBI is a bidirectional link.

DPI

Displays that support Video mode use DPI. The Video mode display device requires a continuous stream of pixel data and temporally accurate synchronization signals to be provided through the DSI link.

The Video mode display has very low control capability. It supports the ShutDown (SD), and ColorMode (CM) commands, and does not support the read commands. Therefore, the Video mode display requires only a unidirectional link, unless any other system interface requires a bidirectional link.

Most of the Video mode displays support very high resolutions and require high bandwidth from the DSI link.

eDPI

The Enhanced DPI supports DCS memory write commands with bandwidth similar to that of DPI. It is an ideal interface for the Command mode displays that have high bandwidth requirements. This interface supports 16/18/24 bpp color depth modes.

The eDPI must be selected if the Command mode display requires pixel data bandwidth greater than 660 Mbps for display refreshment, and is confined to any one of 16, 18, or 24 bpp modes.

Several host systems require the capability to support a broader range of displays, both Video and Command modes. For these systems, the eDPI is an efficient interface, because of the eDPI Standard, and Advanced operational modes, which addresses the Video and Command mode displays respectively, sharing the same resources and interface pins of a native DPI.

2.3 DPI

2.3.1 Overview of DPI

The DPI follows the MIPI DPI-2 specification with pixel data bus width up to 24 bits. In Video mode, it transmits information from the host processor to the peripheral in the form of a real-time pixel stream. This interface allows sending ShutDown (SD) and ColorMode (CM) commands, which directly trigger the interface pins. To transfer additional commands (for example, to initialize the display), an [APB Slave Generic Interface](#) must complement the DPI.



The DPI does not operate concurrently with the DBI.

2.3.2 Enabling DPI

To use the DPI, select DPI or DBI&DPI in coreConsultant while configuring DWC_mipi_dsi_host controller. This interface has a pin that is not included in the DPI-2 specification. The DPI has the `edpihalt` pin, to aid the video driver in controlling the data input rate, when the controller is executing an instruction and cannot accept additional pixel data. For more information about the behavior of `edpihalt`, see ["Transmission of Commands" on page 76](#).

The DPI captures the data and control signals and conveys them to the FIFO interfaces, which subsequently transmit them to the DSI link. Two different data streams presented at the interface are - video control signals and pixel data. Depending on the interface color coding, the pixel data is disposed differently throughout the `dipiidata` bus. [Figure 2-2](#) illustrates the interface pixel color coding.

It is possible to configure the DPI to increase flexibility and promote correct usage of this interface for several systems. These configuration options are as follows:

- Depending on the system requirements, it is possible to program the control signals to change its polarity.
- After reset, DPI waits for the first VSYNC active transition, to start the signal sampling, including the pixel data. This avoids starting the image data transmission in the middle of a frame.
- The number of pixels programmed in the `vid_pkt_size` register field must be a multiple of four, if pixel color coding is 18 bits and the 18-bit loosely packed stream is disabled. This means that, in this mode, the two LSBs in the configuration are always inferred as zero. Additionally, the specification states that, the pixel line size must be a multiple of four.
- To avoid FIFO underflows and overflows, assume that the configured number of pixels are received at all times. This happens even if the `dipiidataen` pin is active for more or less time than necessary. A set of bits in the `VID_PKT_STATUS` register report the status of the FIFOs and internal buffers associated with DPI support.
- Separate the memory space of different video packets using the number of pixels per packet parameter, to keep the memory organized with respect to the packet scheduling.

For `dpishutdn` and `dpicolorm` sampling and transmission, the DPI video signaling must be active. Thus, if the commands do not actively generate the video signals like VSYNC and HSYNC, then these signals are not transmitted through the DSI link. Because of these constraints (and to ensure a flawless transmission of commands), the first VSYNC active pulse must occur for the command sampling and transmission.

Immediately after issuing the command to shutdown the display, DPI video must be kept active for one frame, thus ensuring command transmission.

The DPI-2 specification does not define a standard way to get certain data formats through the pixel interface. To do so, the bit width of the data bus from DPI-2 has been extended from 24-bit to 30-bit and the location of the color components are defined according to [Table 2-1](#) on page 43.

All the uncompressed data types comply with a rate of one pixel per clock cycle, except the 36-bit format and 12-bit YCbCr 4:2:0 format. The exceptions are as follows:

- For 36-bit format, by using the same clock it is not possible to continuously receive one entire pixel per clock cycle, because this data is stored in a 32-bit RAM. Therefore, each clock cycle receives half a pixel, making it {R[11:0], G[5:0]} in one cycle, and the remaining {G[11:6], B[11:0]} of the same pixel in the next one.
- For 12-bit YCbCr 4:2:0, even pixels take 16 bits of information while odd pixels take only 8 bits. It is possible to fit two pixels in 24-bit, transferred in a single clock cycle.

The number of pixels per line must be even for all the YCbCr data types, to meet the number of bytes per packet restrictions, defined in the MIPI DSI specification.

The 30-bit and 36-bit data types are generic and intended to accommodate the data also for non-RGB color spaces in the future.

Figure 2-2 Location of Color Components in an Extension of DPI-2 Pixel Interface

ORDER	20-Bit YCbCr 4:2:2 Loosely Packed		24-Bit YCbCr 4:2:2		16-Bit YCbCr 4:2:2		30-Bit	36-Bit		12-Bit YCbCr 4:2:0		16-Bit Config 1	16-Bit Config 2	16-Bit Config 3	18-Bit Config 1	18-Bit Config 2	24-Bits	DSC24		
	1 st Cycle	2 nd Cycle	1 st Cycle	2 nd Cycle	1 st Cycle	2 nd Cycle		1 st Cycle	2 nd Cycle	Odd line	Even line									
D29							R[9]													
D28							R[8]													
D27							R[7]													
D26							R[6]													
D25							R[5]													
D24							R[4]													
D23	Y1[9]	Y2[9]	Y1[11]	Y2[11]	Y1[7]	Y2[7]	R[3]			Y1[7]	Y1[7]						R[7]	Byte1[7]		
D22	Y1[8]	Y2[8]	Y1[10]	Y2[10]	Y1[6]	Y2[6]	R[2]			Y1[6]	Y1[6]						R[6]	Byte1[6]		
D21	Y1[7]	Y2[7]	Y1[9]	Y2[9]	Y1[5]	Y2[5]	R[1]			Y1[5]	Y1[5]						R[5]	Byte1[5]		
D20	Y1[6]	Y2[6]	Y1[8]	Y2[8]	Y1[4]	Y2[4]	R[0]			Y1[4]	Y1[4]						R[4]	Byte1[4]		
D19	Y1[5]	Y2[5]	Y1[7]	Y2[7]	Y1[3]	Y2[3]	G[9]			Y1[3]	Y1[3]						R[3]	Byte1[3]		
D18	Y1[4]	Y2[4]	Y1[6]	Y2[6]	Y1[2]	Y2[2]	G[8]			Y1[2]	Y1[2]						R[2]	Byte1[2]		
D17	Y1[3]	Y2[3]	Y1[5]	Y2[5]	Y1[1]	Y2[1]	G[7]	R[1]		G[11]	Y1[1]	Y1[1]				R[1]	R[0]	R[5]	R[1]	Byte1[1]
D16	Y1[2]	Y2[2]	Y1[4]	Y2[4]	Y1[0]	Y2[0]	G[6]	R[0]		G[10]	Y1[0]	Y1[0]				R[0]	R[4]	R[0]	R[4]	Byte1[0]
D15	Y1[1]	Y2[1]	Y1[3]	Y2[3]			G[5]	R[9]		G[9]	Y0[7]	Y0[7]	R[4]				R[3]		G[7]	Byte2[7]
D14	Y1[0]	Y2[0]	Y1[2]	Y2[2]			G[4]	R[8]		G[8]	Y0[6]	Y0[6]	R[3]				R[2]		G[6]	Byte2[6]
D13			Y1[1]	Y2[1]			G[3]	R[7]		G[7]	Y0[5]	Y0[5]	R[2]	G[5]	G[5]	R[1]	G[5]	G[5]	Byte2[5]	
D12			Y1[0]	Y2[0]			G[2]	R[6]		G[6]	Y0[4]	Y0[4]	R[1]	G[4]	G[4]	R[0]	G[4]	G[4]	Byte2[4]	
D11	Cb1[9]	Cr2[9]	Cb1[11]	Cr2[11]	Cb1[7]	Cr2[7]	G[1]	R[5]	B[11]	Y0[3]	Y0[3]	R[0]	G[3]	G[3]	G[5]	G[3]	G[3]	G[3]	Byte2[3]	
D10	Cb1[8]	Cr2[8]	Cb1[10]	Cr2[10]	Cb1[6]	Cr2[6]	G[0]	R[4]	B[10]	Y0[2]	Y0[2]	G[5]	G[2]	G[2]	G[4]	G[2]	G[2]	G[2]	Byte2[2]	
D9	Cb1[7]	Cr2[7]	Cb1[9]	Cr2[9]	Cb1[5]	Cr2[5]	B[9]	R[3]	B[9]	Y0[1]	Y0[1]	G[4]	G[1]	G[1]	G[3]	G[1]	G[1]	G[1]	Byte2[1]	
D8	Cb1[6]	Cr2[6]	Cb1[8]	Cr2[8]	Cb1[4]	Cr2[4]	B[8]	R[2]	B[8]	Y0[0]	Y0[0]	G[3]	G[0]	G[0]	G[2]	G[0]	G[0]	G[0]	Byte2[0]	
D7	Cb1[5]	Cr2[5]	Cb1[7]	Cr2[7]	Cb1[3]	Cr2[3]	B[7]	R[1]	B[7]	Cb[7]	Cr[7]	G[2]			G[1]		B[7]	B[7]	Byte3[7]	
D6	Cb1[4]	Cr2[4]	Cb1[6]	Cr2[6]	Cb1[2]	Cr2[2]	B[6]	R[0]	B[6]	Cb[6]	Cr[6]	G[1]			G[0]		B[6]	B[6]	Byte3[6]	
D5	Cb1[3]	Cr2[3]	Cb1[5]	Cr2[5]	Cb1[1]	Cr2[1]	B[5]	G[5]	B[5]	Cb[5]	Cr[5]	G[0]			B[4]	B[5]	B[5]	B[5]	Byte3[5]	
D4	Cb1[2]	Cr2[2]	Cb1[4]	Cr2[4]	Cb1[0]	Cr2[0]	B[4]	G[4]	B[4]	Cb[4]	Cr[4]	B[4]	B[4]	B[4]	B[4]	B[4]	B[4]	B[4]	Byte3[4]	
D3	Cb1[1]	Cr2[1]	Cb1[3]	Cr2[3]			B[3]	G[3]	B[3]	Cb[3]	Cr[3]	B[3]	B[3]	B[3]	B[3]	B[3]	B[3]	B[3]	Byte3[3]	
D2	Cb1[0]	Cr2[0]	Cb1[2]	Cr2[2]			B[2]	G[2]	B[2]	Cb[2]	Cr[2]	B[2]	B[2]	B[2]	B[2]	B[2]	B[2]	B[2]	Byte3[2]	
D1			Cb1[1]	Cr2[1]			B[1]	G[1]	B[1]	Cb[1]	Cr[1]	B[1]	B[1]	B[1]	B[1]	B[1]	B[1]	B[1]	Byte3[1]	
D0			Cb1[0]	Cr2[0]			B[0]	G[0]	B[0]	Cb[0]	Cr[0]	B[0]	B[0]	B[0]	B[0]	B[0]	B[0]	B[0]	Byte3[0]	

According to [Table 2-1](#), the number of payload pixels must be restricted to a multiple of a value. For 30-bit and 36-bit formats, the boundaries of the pixels may not always fill all the bits of the last byte of the payload. In that case, the controller automatically fills the remaining bits with 0.

Table 2-1 Multiplicity of the Payload Size in Pixels for Each Data Type

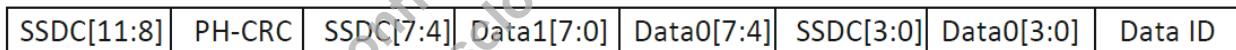
Value	Data Types
1	<ul style="list-style-type: none"> ■ 16-bit ■ 18-bit loosely packed ■ 24-bit ■ 30-bit ■ 36-bit ■ DSC24 compressed data
2	<ul style="list-style-type: none"> ■ Loosely Packed Pixel Stream, 20-bit YCbCr 4:2:2 ■ 24-bit YCbCr 4:2:2 ■ 16-bit YCbCr 4:2:2 ■ 12-bit YCbCr 4:2:0
4	<ul style="list-style-type: none"> ■ 18-bit non-loosely packed

2.3.3 Configuring the Display Pixel Interface

For more information, see the “Configuring the Display Pixel Interface” section of the *DWC MIPI DSI Host Controller User Guide*.

2.3.4 Video Mode with Stereoscopic Image Data

Figure 2-3 illustrates the bytes in a short packet.

Figure 2-3 Bytes in a Short Packet**Figure 2-4** Bytes in a Short Packet (C-PHY)

On writing the SDF_3D register, it is possible to send the next VSS packet, if necessary, as listed in Table 2-2. The controller sends this packet, with data 0 byte set to 0x08, as bit 3 indicates the presence of 3D control payload in data 1 byte (the other bits of data 0 are reserved). The data 1 includes a copy of SDF_3D[5:0], the 3D control configuration.

Table 2-2 VSS with 3D Information

3D was Active (SDF_3D[1:0] was 2'b10 or 2'b01)	3D Activated (SDF_3D[1:0] switched to 2'b10 or 2'b01)	SDF_3D Updated	send_3d_cfg (SDF_3D [16])	Next VSS with 3D Information
*	*	0	0	False
False	False	1	0	False
False	True	1	0	True

3D was Active (SDF_3D[1:0] was 2'b10 or 2'b01)	3D Activated (SDF_3D[1:0] switched to 2'b10 or 2'b01)	SDF_3D Updated	send_3d_cfg (SDF_3D [16])	Next VSS with 3D Information
True	False	1	0	True
True	True	1	0	True
*	*	*	1	True

The 3D control also indicates that the 2D mode is on. When switching from a 3D mode to 2D (SDF_3D[1:0]=2'b00), the next VSS packet contains data 0 set to 0x08 and data 1 has the two LSB set to '00' to indicate that the 3D and the 2D mode are off.

Setting the SDF_3D[16] bit, sends the 3D control configuration information in every VSS packet. If not, the VSS with 3D information is only sent in the next frame.

On updating the SDF_3D register with 3D disabled and without activating the 3D information, then the next VSS does not have the 3D information. This behavior ensures compatibility with devices that do not understand this packet.

2.3.5 Video Transmission Modes

The different video transmission modes are as follows:

- Burst mode
- Non-Burst mode
 - Non-Burst mode with sync pulse
 - Non-Burst mode with sync event

2.3.5.1 Burst Mode

In this mode, the controller buffers the entire active pixel line into a FIFO and transmits it in a single packet with no interruptions. This transmission mode requires DPI Pixel FIFO to have the capacity to store a full line of active pixel data. For optimal performance, there must be a relatively big difference between the pixel required bandwidth and the DSI link bandwidth. This allows the DWC_mipi_dsi_host controller to quickly dispatch an entire active video line in a single data burst and then return to low-power mode.

2.3.5.2 Non-Burst Mode

In this mode, the processor uses the partitioning properties of the DWC_mipi_dsi_host controller to divide the video line transmission into several DSI packets, so that the pixel required bandwidth matches with the DSI link bandwidth. By doing so, the controller does not require a full line of pixel data to be stored inside the DPI Pixel FIFO, but instead it requires only one video packet content to be stored.

2.3.5.3 Guidelines for Selecting the Burst or Non-Burst Mode

Selecting the Burst or Non-Burst mode depends on system configuration and device requirements. Choose the video transmission mode that suits the application scenario. The Burst mode is beneficial because it

increases the probability of the link spending more time in the low-power mode, thereby decreasing power consumption. However, the following conditions must be met for availing the maximum benefits from the Burst mode of operation:

- The DWC_mipi_dsi_host controller must have sufficient memory to store a pixel line without causing FIFOs overflow.
- The display device must be able to receive a pixel line in a single burst without causing a reception buffer overflow.
- The DSI output bandwidth must be higher than the DPI system interface input bandwidth, in such a way, that enables the link to go to low-power once per line.

In Burst mode, if the system does not meet the above requirements, then it can result in display device malfunctioning due to pixel data loss. This error possibility depends on the system capability to store temporary pixel data.

If all the conditions for using the Burst mode cannot be met, use the Non-Burst mode to avoid the errors caused by the Burst mode. The Non-Burst mode provides a better matching of rates for pixel transmission, enabling:

- Only a certain amount of pixels to be stored in the memory and not an entire pixel line (lesser DPI RAM requirements in the DWC_mipi_dsi_host controller)
- Operation with devices that support only a small amount of pixel buffering (less than a pixel line)

The DSI Non-Burst mode must be configured in such way that DSI's output pixel ratio matches DPI's input pixel ratio, reducing the memory requirements on both host and device sides, or on either host or device side. Dividing a pixel line into several chunks of pixels and optionally interleaving them with null packets helps in achieving this.

The following equations describe the programming of the transmission parameters of the DWC_mipi_dsi_host controller in Non-Burst mode, to match DSI's pixel output ratio (left hand side of the "==" sign) and DPI's pixel input ratio (right hand side of the "==" sign).

- On enabling the null packets, program the transmission parameters as follows:

$$((\text{lanebyteclkperiod} * \text{vid_num_chunks} * \text{vid_pkt_size} * \text{bytes_per_pixel} + 2 * \text{long_packet_overhead} + \text{vid_null_size}) / \text{number_of_lanes} * \text{word_length}) = \text{pixels_per_line} * \text{dpipclkperiod}$$

- On disabling the null packets, program the transmission parameters as follows:

$$((\text{lanebyteclkperiod} * \text{vid_num_chunks} * \text{vid_pkt_size} * \text{bytes_per_pixel} + \text{long_packet_overhead}) / \text{number_of_lanes} * \text{word_length}) = \text{pixels_per_line} * \text{dpipclkperiod}$$

For D-PHY, the long_packet_overhead is 6 and the word_length is 1. For C-PHY, the long_packet_overhead is (14 + 6 * number_of_lanes), and the word_length is 2.

2.3.6 Updating DPI Configuration Dynamically

Using shadow registers, it is possible to update DPI configuration dynamically without impacting the current frame. The VID_SHADOW_CTRL register controls this feature.

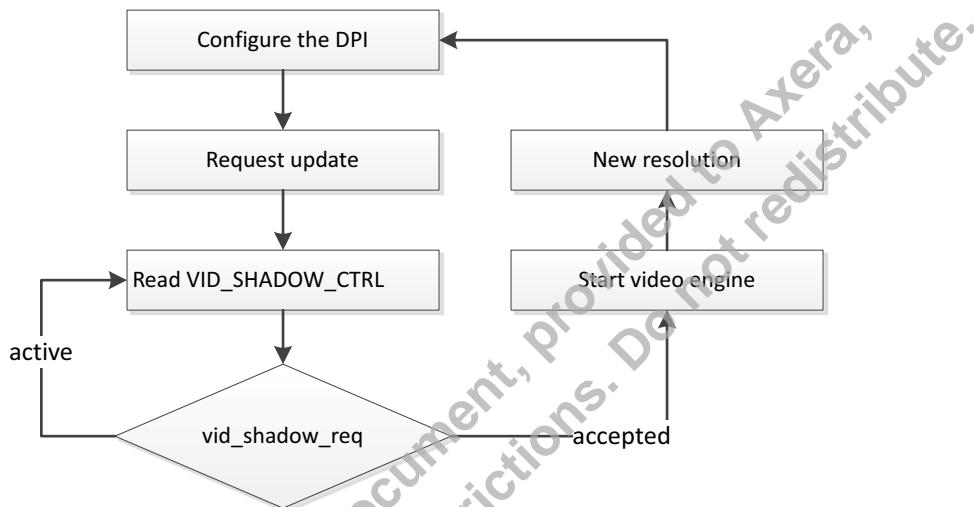
The controller uses the new configuration, only when the system requests it.

To update DPI configuration during the transmission of a video frame, the configuration of that frame must be stored in auxiliary registers. This allows the new frame configurations to be set through the APB interface without corrupting the current frame.

By default, this feature is disabled. To enable this feature, set the `vid_shadow_en` field of the `VID_SHADOW_CTRL` register to 1. On enabling this feature, the system provides the configuration in the auxiliary registers.

Figure 2-5 illustrates the necessary steps to update the DPI configuration.

Figure 2-5 Flow to Update the DPI Configuration using Shadow Registers

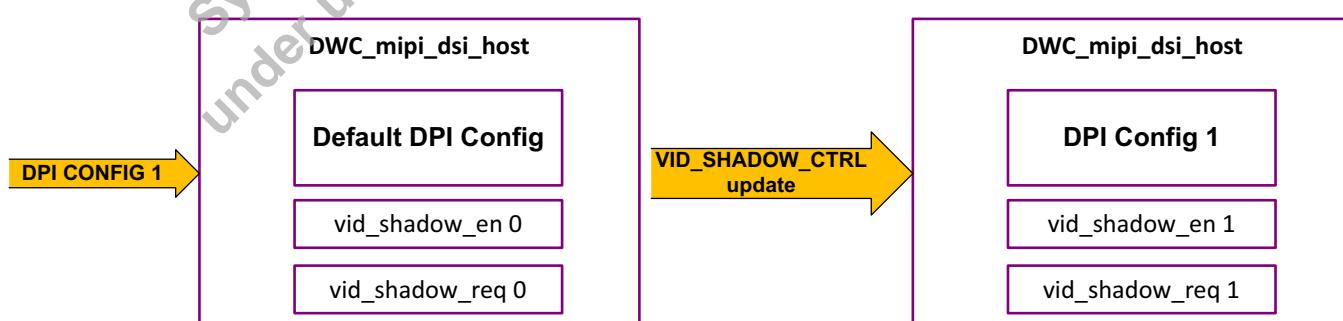


2.3.6.1 Immediate Update

On enabling the shadow register feature, the auxiliary registers require DPI to be configured before the video engine starts. This means that, after a reset, `vid_shadow_req` is immediately granted.

To immediately update the active registers without the reset, set the `vid_shadow_en` and `vid_shadow_req` bits of the `VID_SHADOW_CTRL` register to 0.

Figure 2-6 Immediate Update Procedure

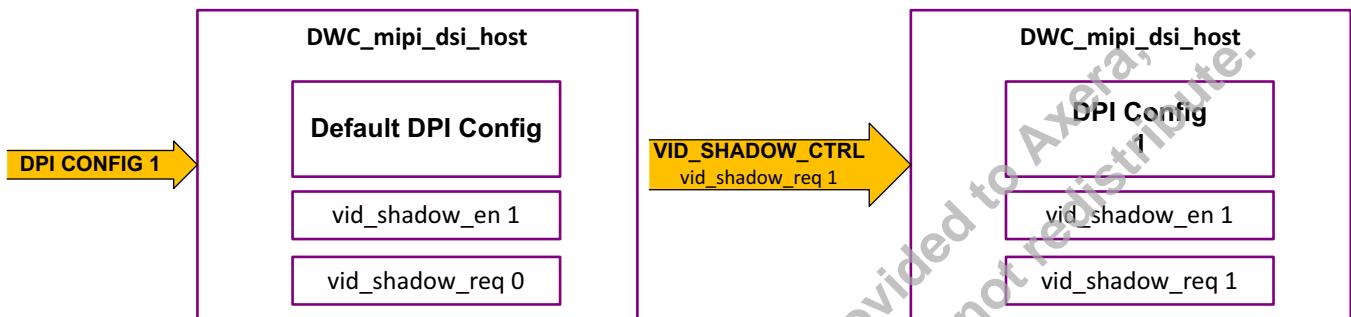


2.3.6.2 Updating the Configuration During the Transmission of a Frame Using APB

To update the DPI configuration, follow these steps:

1. Set the vid_shadow_en bit of the VID_SHADOW_CTRL register to 1.
2. Set the vid_shadow_req bit of VID_SHADOW_CTRL register to 1.
3. Monitor the vid_shadow_req bit. Update completion sets this bit to 0.

Figure 2-7 Update the Configuration During the Transmission of a Frame

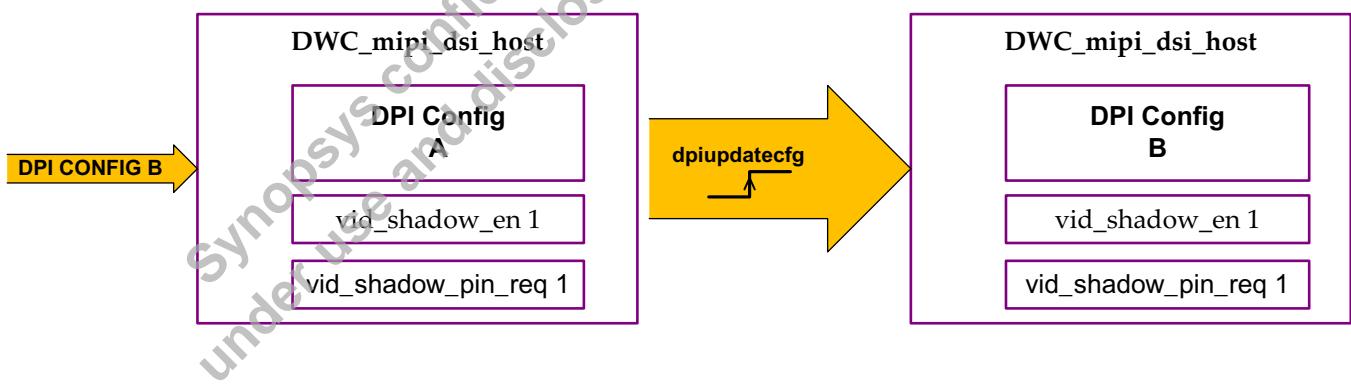


2.3.6.3 Updating the Configuration During the Transmission of a Frame Using the External Signal

If video engine's resolution change request is not possible to go through APB, use the dpiupdatecfg pin to request it. To monitor this pin, set the vid_shadow_pin_req bit of the VID_SHADOW_CTRL register to 1.

At the rising edge of the dpiupdatecfg signal, an update request is generated.

Figure 2-8 Update Configuration Using the External Signal



2.3.6.4 Requesting a Configuration Update

It is possible to request for the DPI configuration update at any part of the frame, because the DWC_mipi_dsi_host controller waits until the end of the frame to update it. However, avoid sending the update request during the first line of the frame because the data must propagate between clock domains.

2.3.7 Signals Related to DPI

For a description of signals related to DPI, see “Signal Descriptions” chapter.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

2.3.8 Registers Related to DPI

Registers related to DPI are as follows:

- DPI_VCID
- DPI_COLOR_CODING
- DPI_CFG_POL
- DPI_LP_CMD_TI

For more information, see “Register Descriptions” chapter.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

2.4 eDPI

2.4.1 Overview of eDPI

The eDPI is an enhanced DPI with an additional functional mode. Based on the MIPI DPI pin description, this additional mode enables the following:

- DCS memory write commands transmission
- Reusing DPI pinout
- Pixel color mapping

This interface has the following modes of operation:

- Standard Video mode (DPI standard interface)
- Adapted Command mode (Synopsys proprietary interface)

The Standard Video mode of operation sets the interface to operate as a native DPI. For more information about this mode of operation, see “[DPI](#)” on page [41](#).

2.4.2 Enabling eDPI

To enable Enhanced DPI, select Enhanced DPI in the coreConsultant while configuring the `DWC_mipi_dsi_host` controller.



Note The eDPI does not operate concurrently with DBI and DPI, or with either DBI or DPI.

2.4.3 Description of eDPI

The eDPI, operating in Adapted Command mode, enables the system to input a pixel information stream conveyed by the controller using the Command mode transmission (using the DCS packets).

The eDPI also supports the following:

- Pixel input control rate signaling (edpihalt)
- Tearing effect report mechanism (edpite)

The Adapted Command mode allows sending a large amount of data through the following:

- `memory_write_start` (WMS)
- `memory_write_continue` (WMC) DCS commands

It provides a wider data transfer bandwidth for memory write operations sent in Command mode to MIPI display, while enabling the process of refreshing large pixels areas in high resolution displays. To transfer additional commands, such as display configuration commands, read back commands, and tearing effect initialization, the [APB Slave Generic Interface](#) must be used to complement the eDPI.

Selecting eDPI in the coreConsultant, selects the generic packets support in the configuration menu. This is the requirement because the eDPI supports only WMS and WMC commands. The command mode display requires other commands for proper operation.

In the Adapted Command mode, the eDPI captures data and control signals. The eDPI then conveys these signals to the FIFO interfaces, to allow its transmission to the DSI link, by reusing DPI's FIFOs to store the payload data for WMS and WMC commands. [Figure 2-9](#) illustrates this.

The data streams present at the interface are as follows:

- Video control signals
- Pixel data

A set of bits in the VID_PKT_STATUS register reports the status of the FIFOs and internal buffers associated with eDPI support.

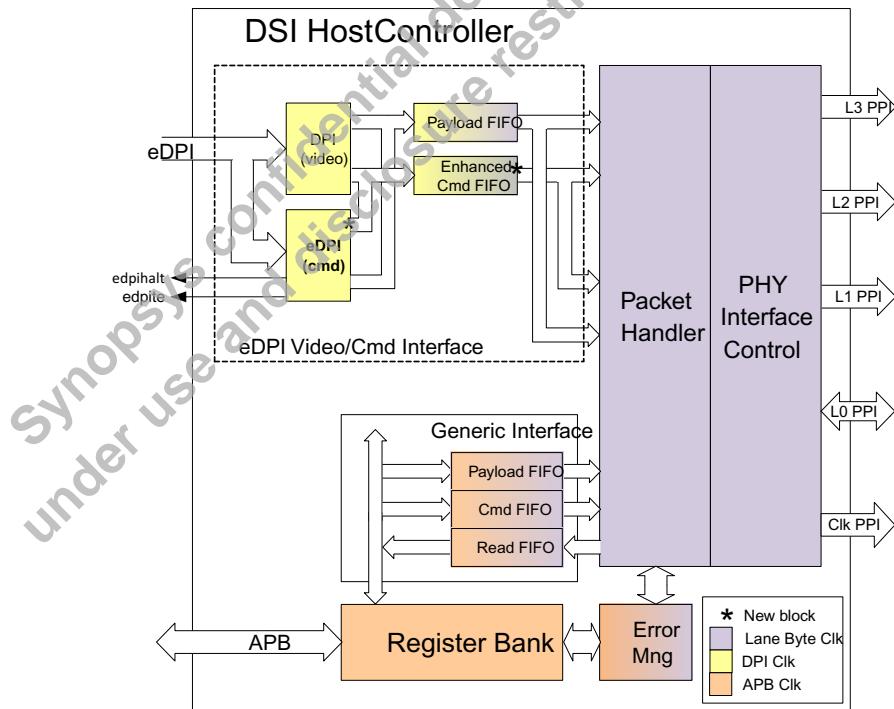
In the Adapted Command mode of operation, eDPI supports:

- 16 bpp
- 18 bpp
- 24 bpp RGB
- DSC24 compressed data

In the standard Video mode of operation, eDPI provides the following additional support:

- YCbCr modes
- 30 bpp
- 36 bpp RGB

Figure 2-9 eDPI Block Diagram



To transmit image data in Command mode through the eDPI, follow the below steps:

- Use the set_column_address and set_page_address DCS commands, to define the image area to be refreshed. Define the image area once (this value remains effective until further update).
- Use the dpi_color_coding field of the DPI_COLOR_CODING register, to define the pixel color coding. Use the dpi_vc_id field of the DPI_VC_ID register, to define the eDPI generated packets' virtual channel. These need to be defined only once.
- Start eDPI data transmission.

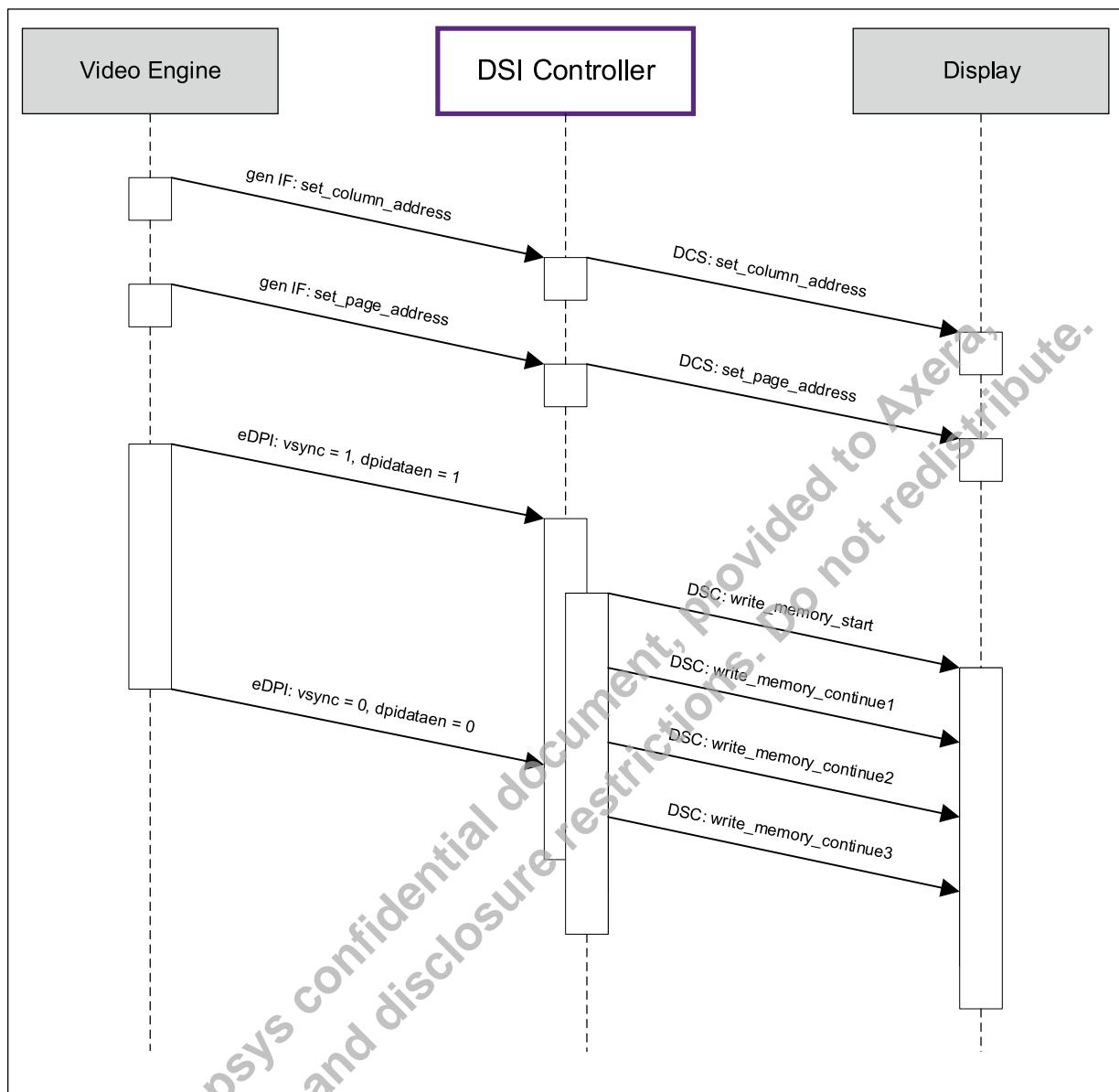
To transmit image data in Command mode using the eDPI:

- set cmd_video_mode bit of the MODE_CFG register to 1

To transmit image data in Video mode using the eDPI:

- set cmd_video_mode bit of the MODE_CFG register to 0

Figure 2-10 on page 53 shows the eDPI usage flow.

Figure 2-10 eDPI Usage Flow

2.4.4 eDPI in Adapted Command Mode

On setting the cmd_video_mode bit of the MODE_CFG register to 1, the eDPI pins assume the behavior corresponding to the Adapted Command mode.

In this mode, the host processor uses the eDPI to transmit a continuous pixel stream, to be written in the local frame buffer of the peripheral.

It uses a pixel input bus to receive the pixels and dpivsync_edpiwms control signal, to limit the stream of continuous pixels.

When the dpivsync_edpiwms signal rises, the current value of the edpi_allowed_cmd_size field of the EDPI_CMD_SIZE register, is shadowed to the internal interface function. While this signal is high, the interface increments a counter on every valid pixel input through the interface. When this pixel counter

reaches `edpi_allowed_cmd_size`, the controller writes a command into the command FIFO and the packet is ready to be transmitted through the DSI link.

If the `dpiivsync_edpiwms` signal falls before the counter reaches the value of shadowed `edpi_allowed_cmd_size`, the controller sends a WMS command to the command FIFO, with word count (WC) set to the amount of bytes that correspond to the value of the counter. On receiving (shadowed value) more than `edpi_allowed_cmd_size` number of pixels, the controller sends a WMS command to the command FIFO with WC set to the number of bytes that correspond to `edpi_allowed_cmd_size` and restarts the counter.



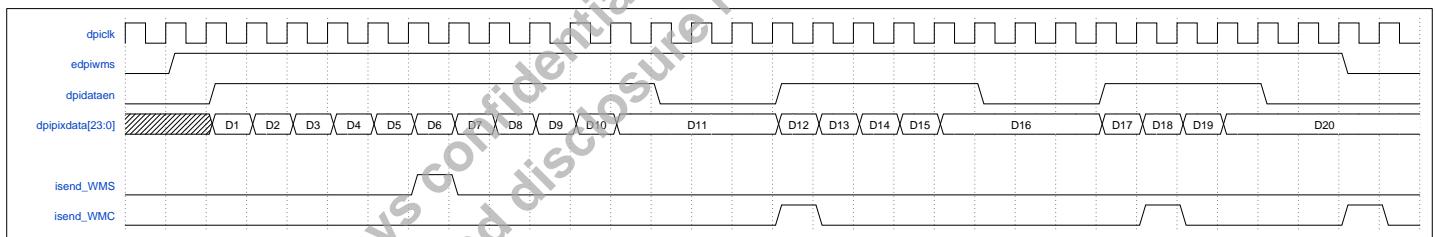
Note The value of `edpi_allowed_cmd_size` is defined in pixels and not in bytes as the WC packet field. The DWC_mipi_dsi_host controller automatically converts the pixels to bytes.

On writing the first WMS command to the FIFO, the circuit behaves in a similar way, but issues WMC commands instead of WMS commands. The process is repeated until `dpiivsync_edpiwms` falls. The controller automatically starts sending a new packet when `dpiivsync_edpiwms` falls or on reaching the `edpi_allowed_cmd_size` limit.

The payload FIFOs always operate according to `dpidataen` signal. This means that, it is possible to write the FIFOs in chunks, if necessary. [Figure 2-11](#) shows an example in which the value of `edpi_allowed_cmd_size` corresponds to six words of 24 bits per pixel.

After `dpiivsync_edpiwms` rises, a new WMS command is scheduled. When six words have been received, a WMS command is written to the command FIFO. After this, each time six words are stored, a new WMC is sent and scheduled until `dpiivsync_edpiwms` falls. When this happens, if the pixel counter is not 0, one last WMC command is written with the remaining payload that is in the payload FIFO.

Figure 2-11 WMS and WMC Commands in Automatic Partitioning Mode



2.4.5 Support for Tearing Effect (TE)

The DSI specification supports tearing effect function in Command mode displays. It enables the host processor to receive timing accurate information about where the display peripheral is, in the frame buffer content reading process.

Configure the following registers, to activate the tearing effect in the controller:

- `CMD_MODE_CFG[0]`: `tear_fx_en`
- `PCKHDL_CFG[2]`: `bta_en`

After TE request, the controller performs a double Bus Turn-Around (BTA) granting the ownership of the link to the DSI display. The display holds the ownership of the bus until the tear event occurs, which is indicated to the controller by a PHY trigger event. Then the DWC_mipi_dsi_host controller decodes the trigger and indicates the event by pulsing the `edpite` signal during one `dpiclk` cycle.

The DWC_mipi_dsi_host controller requests TE from the display module, using the following possibilities:

- By software: TE request using APB interface
- By hardware: TE request using dedicated input pin (tear_request).

2.4.5.1 TE Request by Software

To activate TE request by software set the following register to 0:

- EDPI_TE_HW_CFG [0]: hw_tear_effect_on

To use this function, issue a set_tear_on or set_tear_scanline command after the display update, using the WMS and WMC DCS commands. This procedure halts the DSI link until the display is ready to receive a new frame update.

The DWC_mipi_dsi_host controller does not automatically generate the tearing effect request (double BTA) after a WMS or WMC sequence for flexibility purposes. In this way, several regions of the display can be updated, improving DSI bandwidth usage. Tearing effect request must always be triggered by a set_tear_on or set_tear_scanline command in the DWC_mipi_dsi_host controller implementation.

2.4.5.2 TE Request by Hardware

To activate TE request by the hardware, set the following register to 1.

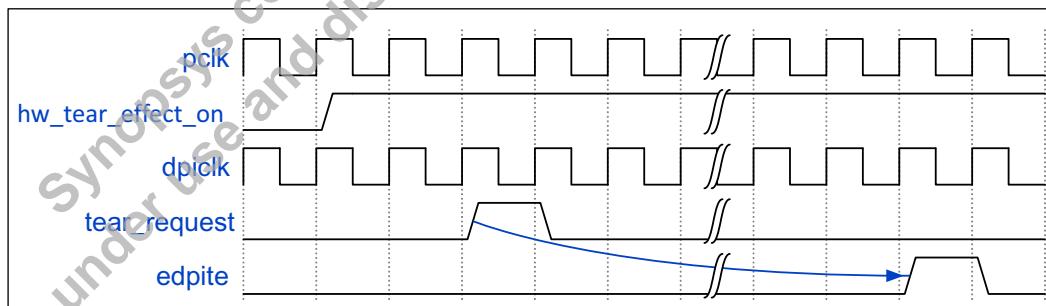
- EDPI_TE_HW_CFG [0]: hw_tear_effect_on

As soon as the bus is free the controller performs a double Bus Turn-Around (BTA) if:

- The feature is active by register bank (hw_tear_effect_on = 1)
- There is also a TE request (tear_request=1).

The controller then decodes the trigger and indicates the event, by pulsing edpите signal during one dpipclk cycle. [Figure 2-12](#) illustrates this.

Figure 2-12 Tearing Effect Request by Hardware



After the assertion of tear_request, the DWC_mipi_dsi_host controller sends a set_tear_on or set_tear_scan_line command to start the tear effect request process. The DWC_mipi_dsi_host controller issues the DSC packet types selected by the EDPI_TE_HW_CFG[1] register:

- set_tear_scan_line - EDPI_TE_HW_CFG[1] register field = 1 (in this case, programming EDPI_TE_HW_CFG[31 : 16] register fields, selects the parameter that describes the Tearing Effect Output Line mode.)

- set_tear_on - EDPI_TE_HW_CFG[1] register field = 0

The DWC_mipi_dsi_host controller ignores the tearing effect request (tear_request=1) if the feature is not active (hw_tear_effect_on = 0). In this situation, it triggers an interrupt to inform the system about the occurrence of the errors.

2.4.5.2.1 Generic Traffic Priorities

In tearing effect performed by the hardware, after tear_request assertion, the EDPI_TE_HW_CFG[1] register field controls the options available for the DWC_mipi_dsi_host controller to start the double BTA:

- EDPI_TE_HW_CFG[1] register field = 0
 - The bus is considered free immediately after transmitting the eDPI traffic, even though there is pending generic traffic.
- EDPI_TE_HW_CFG[1] register field = 1
 - The bus is considered free, after transmitting both eDPI and generic traffic.

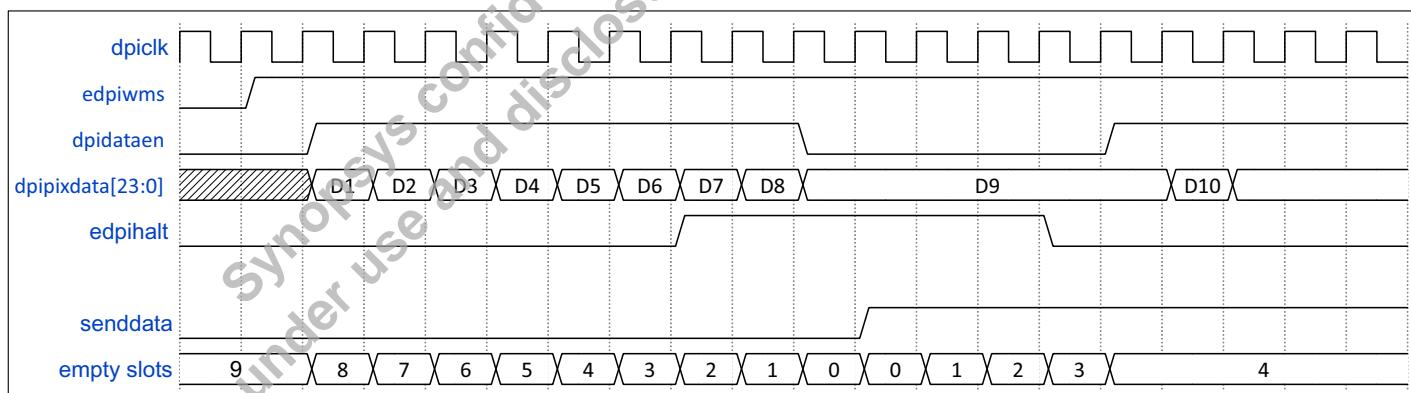
2.4.6 Halt Functionality in Adapted Command Mode

The edpihalt output signal rises each time when either the payload or the command FIFO is about to become full after two clock cycles, and falls in the opposite condition. On asserting the edpihalt signal, dpivsync_edpiwms signal goes low and then high, signaling the end and the start of a new command.

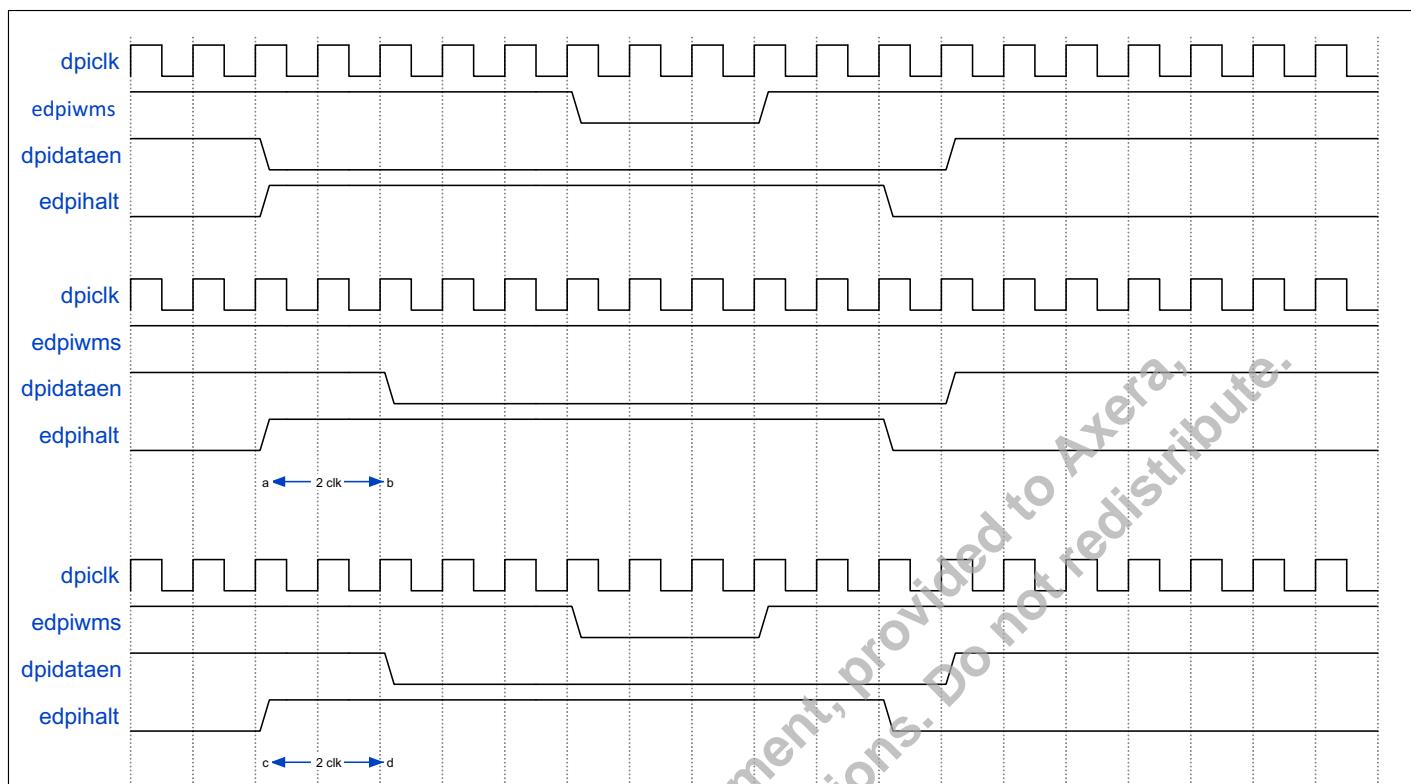
However, during the edpihalt assertion, if the dpivsync_edpiwms signal goes low and then high, then edpihalt must be de-asserted, to assert dpidataen after the rising edge of dpivsync_edpiwms signal. In this case (edpihalt asserted), the two extra dpidataen pulses can occur only before the dpivsync_edpiwms signal goes low.

[Figure 2-13](#) shows the waveform for a sample related to edpihalt signal. In this case, the system has two additional clock cycles of latency to react to edpihalt signal.

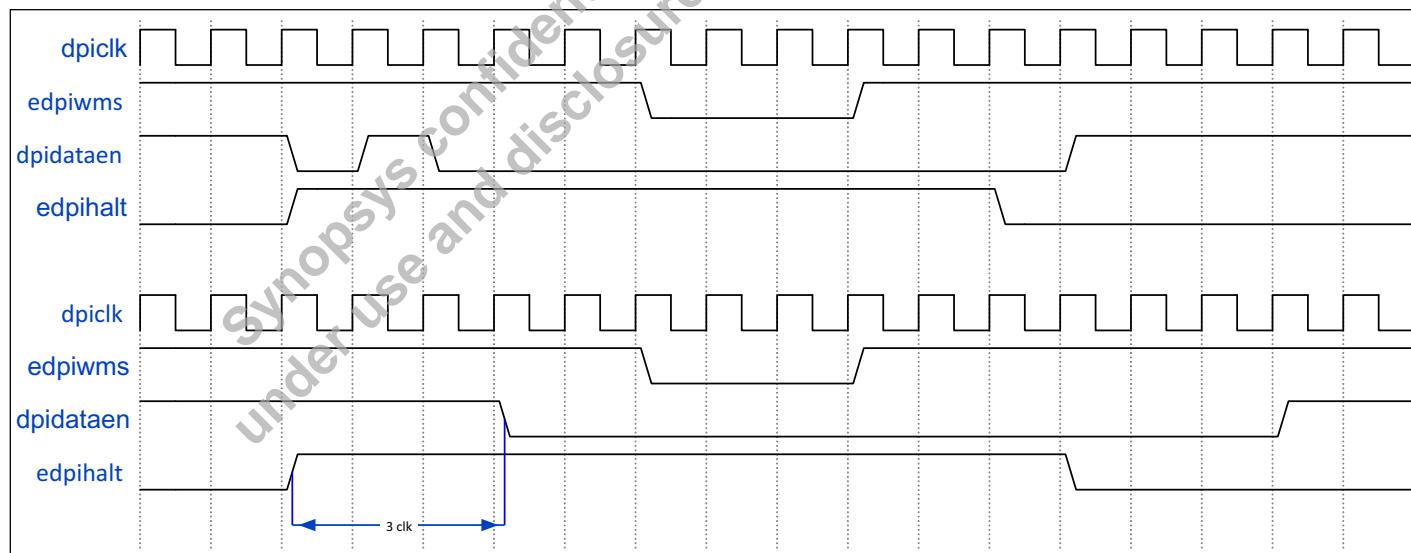
Figure 2-13 Halt Functionality for a System Delay of Two Clock Cycles



[Figure 2-14](#) shows some valid cases related to edpihalt and dpidataen signals with edpihalt asserted.

Figure 2-14 Valid Cases of Halt Functionality and dpidataen

[Figure 2-15](#) shows some invalid cases related to edpihalt and dpidataen signals with edpihalt asserted.

Figure 2-15 Invalid Cases of Halt Functionality and dpidataen

The value programmed in the `edpi_allowed_cmd_size` field of the `EDPI_CMD_SIZE` register is shadowed each time `dpiwms` rises and remains valid until the next time `dpiwms`

rises again. This means that, even if the value in the register bank changes, the initial value is still effective and used along the sequence of WMS and WMC commands.

Therefore, it is possible to set the value of the `edpi_allowed_cmd_size` field of the `EDPI_CMD_SIZE` register in advance, while the previous sequence of commands is being transmitted. This ensures that the intended value is shadowed by the time `dpivsync_edpiwms` rises.

Because the command FIFO has a fixed depth of four words, on setting the `edpi_allowed_cmd_size` field to a very small value, then on sending three commands to this FIFO, the `edpihalt` signal rises soon, even if the payload FIFO is close to empty.

For that reason, to optimize controller utilization, choose a value for `edpi_allowed_cmd_size` such that both FIFOs become nearly full more or less at the same time.

```
edpi_allowed_cmd_size_option = ((pld_fifo_depth-2)*(4/bytes_per_pixel))*(1/3)
```

To ensure proper synchronization of signals between different clock domains, the `dpipclk` signal must remain active at all times.

[Figure 2-16](#) shows the eDPI usage, on receiving the payload in chunks.

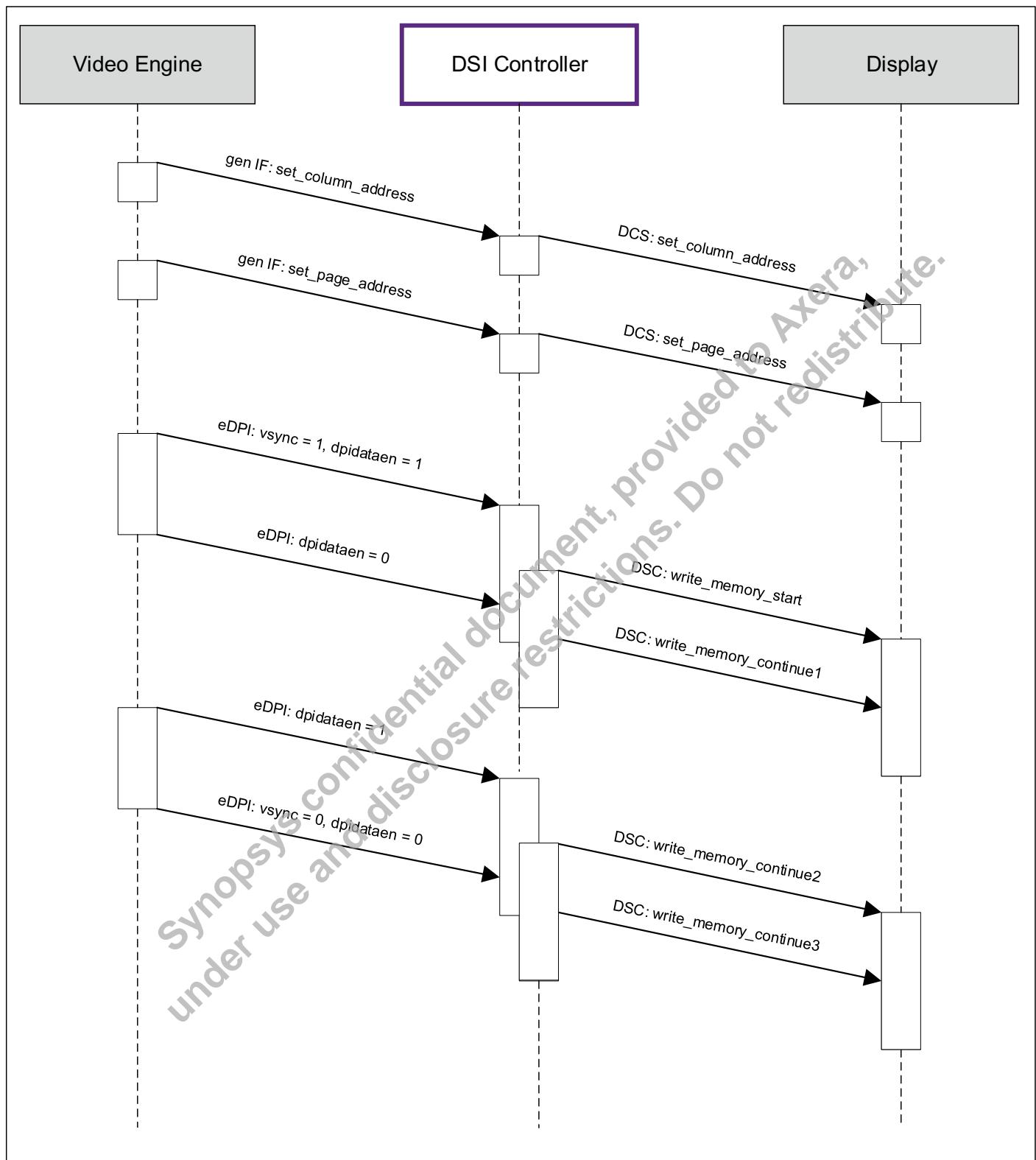
Figure 2-16 Flow when Payload is Received in Chunks

Figure 2-17 shows the eDPI usage when the video engine is faster than the DSI Link.

Figure 2-17 Flow of eDPI when Video Engine Faster than the DSI Link

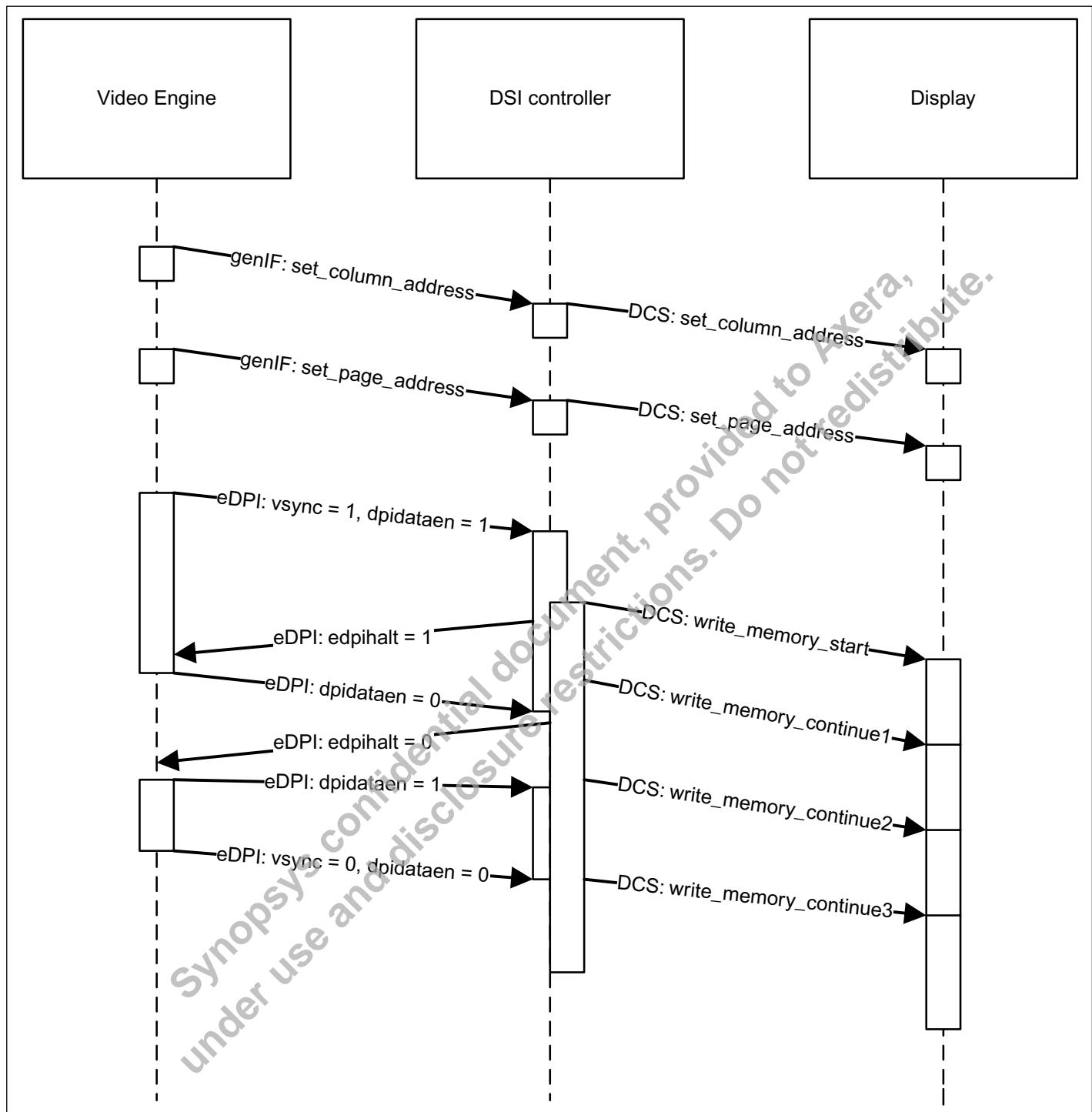
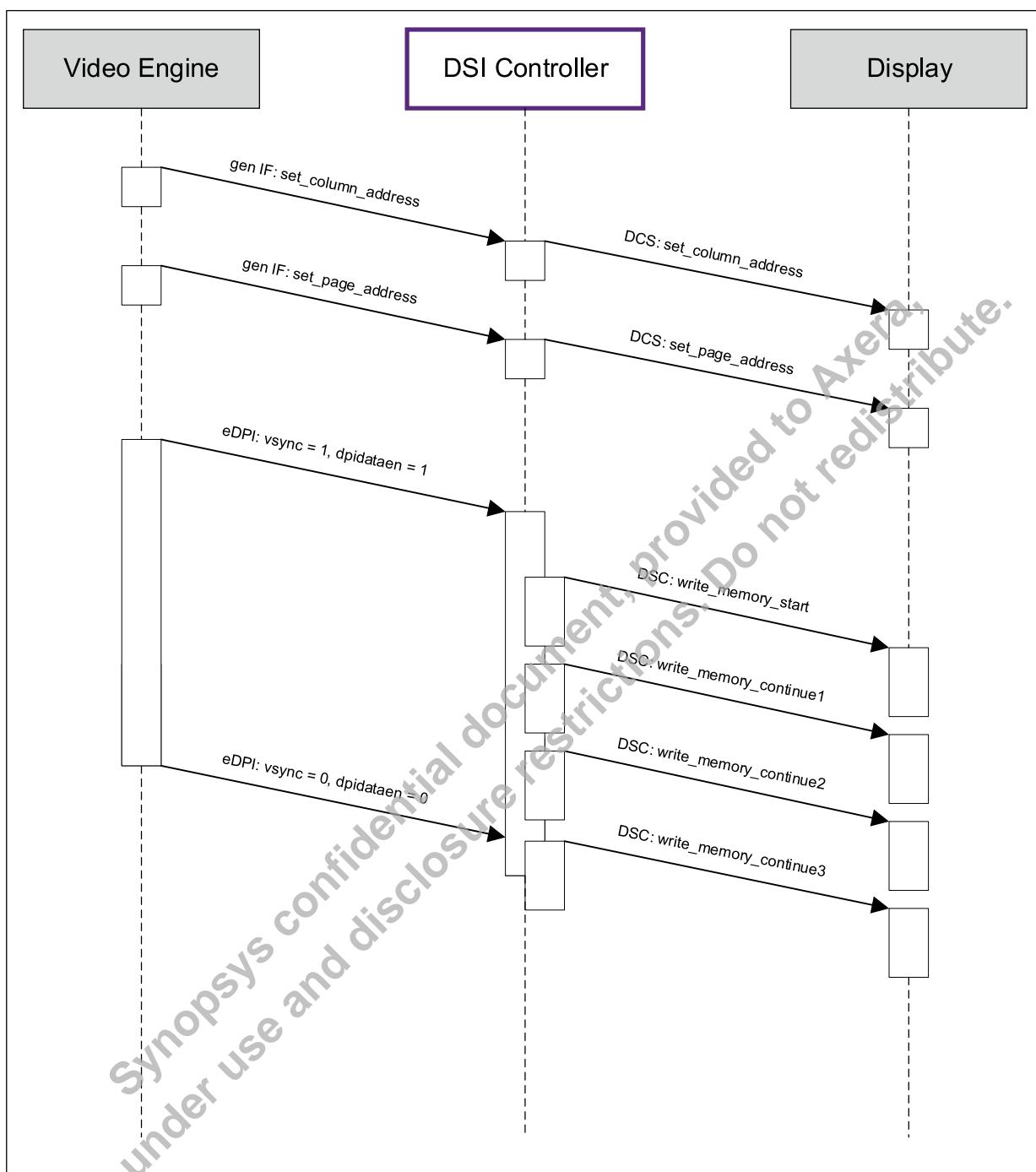


Figure 2-18 shows the eDPI usage when the video engine is slower than the DSI Link.

Figure 2-18 Flow when Video Engine is Slower than the DSI Link

2.4.7 Auto ULPS Control Scheme

It is possible for the `DWC_mipi_dsi_host` controller, to automatically enter into the ultra low-power state mode and turn off the PLL, if required. When eDPI is in Adapted Command mode, the `DWC_mipi_dsi_host` controller evaluates the possibility of entering into ULPS mode without software intervention. To enable this feature, set `auto_ulps` bit of the `AUTO_ULPS_MODE` register to 1.

While using D-PHY, consider the DWC_mipi_dsi_host controller to be in ULPS when:

- Non-continuous clock mode: clock and data lanes are in ULPS
- Continuous clock mode: data lanes are in ULPS

While using C-PHY, consider the DWC_mipi_dsi_host controller to be in ULPS when:

- In the non-continuous clock mode, all data lanes are in ULPS
- In the continuous clock mode:
 - DSI does not enter ULPS mode for a C-PHY system which consists of one data lane.
 - For a C-PHY system which consists of more than one data lane, consider the controller to be in ULPS when all lanes (except data lane 1) are in ULPS mode.

Consider the DWC_mipi_dsi_host controller to be in idle state when:

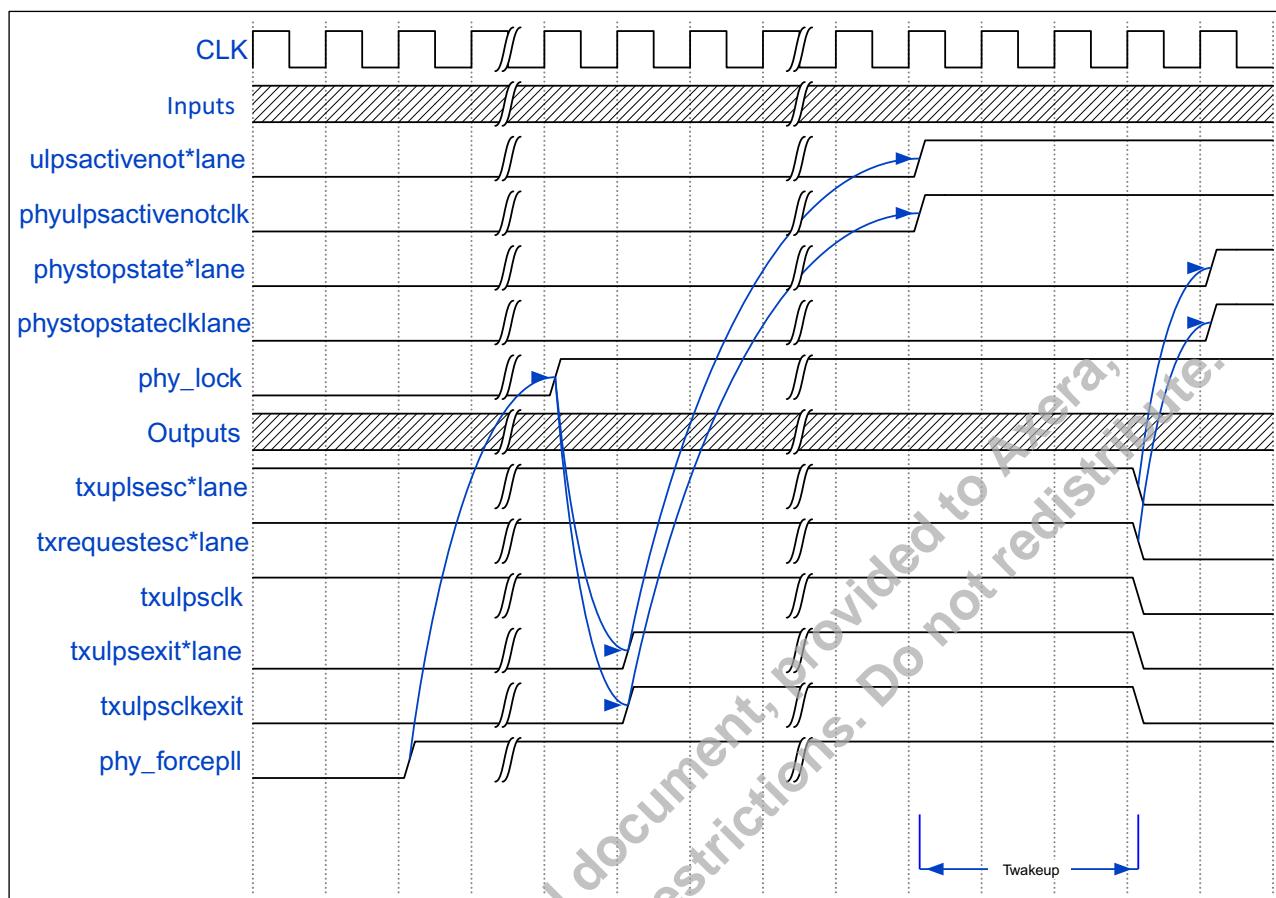
- No APB slave generic pending
- No eDPI traffic is being transmitted
- dpivsync_edpiwms is de-asserted

When in idle state, for the period defined in the ulps_entry_delay field of the AUTO_ULPS_ENTRY_DELAY register, the DWC_mipi_dsi_host controller initiates the ULPS entry sequence. No pending TE on queue or executing (tear effect requires double BTA).

On reaching the ULPS state, the DWC_mipi_dsi_host controller automatically disables the PLL, using the phyforcepll pin, if pll_off_ulps bit of the AUTO_ULPS_MODE register is 1.

Upon reception of an APB slave generic command TE request, or dpivsync_edpiwms assertion, the DWC_mipi_dsi_host controller:

- Turns on the PLL, waits for phy_lock, and initiates the ULPS exit sequence, if pll_off_ulps bit of the AUTO_ULPS_MODE register is 1.
- Initiates the ULPS exit sequence, if pll_off_ulps bit of the AUTO_ULPS_MODE register is 0.

Figure 2-19 Example of ULPS Exit Sequence with D-PHY

As part of ULPS exit sequence, the DWC_mipi_dsi_host controller provides a programmable timer. The AUTO_ULPS_WAKEUP_TIME register specifies the minimum ULPS wake-up time (Twakeup).

$$T_{\text{wakeup}} = t_{\text{twakeup_clk_div}} * t_{\text{twakeup_cnt}}$$

The Timer's granularity ranges between any appropriate values considered by the implementer.

The default Twakeup is 1 ms, as specified by MIPI PHY specifications.

2.4.8 Signals Related to eDPI

For more information of signals related to eDPI, see “Signal Descriptions” chapter.

2.4.9 Registers Related to eDPI

For more information of registers related to eDPI, see “Register Descriptions” chapter.

2.5 DBI

2.5.1 Overview of DBI

The DBI follows the MIPI DBI specification. It transmits the information in Command mode. Here, the transactions primarily take the form of sending the commands defined in the DCS Specification to a peripheral, such as a display module that incorporates a display controller. The DWC_mipi_dsi_host controller supports type A and type B variants of this interface. The DBI supports the Command mode devices where the transactions are carried out through the commands as defined in the DCS Specification.



The DBI does not operate concurrently with DPI.

The DBI encapsulates DCS commands in the DSI packets to be transmitted through the PHY link. Some commands require a response from the device, and the interface provides read data from the device or peripheral.

The DBI has certain limitations on the clock frequency and the data bus dimensions. The maximum bandwidth that can be implemented in the DWC_mipi_dsi_host controller is 660 Mbps, assuming that the DSI D-PHY link operates at 1 Gbps. Therefore, the DBI can be selected only if the display supports Command mode and requires less than 660 Mbps of pixel data bandwidth for refreshing the display.

2.5.2 Enabling DBI

To use the DBI, select DBI or DBI&DPI in the Select the system interface option in coreConsultant.

Use Select the DBI type option, to select any one of the following DBI:

- Type A interface Fixed E Mode
- Type A interface Clocked E Mode
- Type B interface

The controller does not support Type C serial mode. The selection type maps the related pins on the controller's pinout. The DBI uses RGB additive color mixing method, and selects the pixel data format through the DBI_CFG register. The `in_dbi_conf` field defines the input pixel data format, and the `out_dbi_conf` field defines the output pixel data format that is determined by the display device. This interface must be associated with the DSI Virtual Channel configured in the `dbi_vc_id` field.

For more information about diagrams on pixel-to-byte conversion for each mode, see [Appendix B, "DBI Color Code Mapping Waveforms"](#).

2.5.3 DSI Packet Types

The DBI receives commands and decodes information required for command packetizing format.

Depending on the command type, the DWC_mipi_dsi_host controller outputs the related DSI packet type (DCS short write packet, DCS read packet, or DCS long write) according to information present on *MIPI® Alliance Specification for Display Command Set (DCS) Version 1.3*. If the DCS commands have variable sizes, configure the sizes using the `DBI_CMDSIZE` register. The `DBI_CMDSIZE` register must be configured only when the DWC_mipi_dsi_host controller is idle and not transmitting. Commands with zero or one parameter are encapsulated in short packets and commands with more than one parameter are encapsulated in long packets.

2.5.4 Configuring the DBI Pixel Data

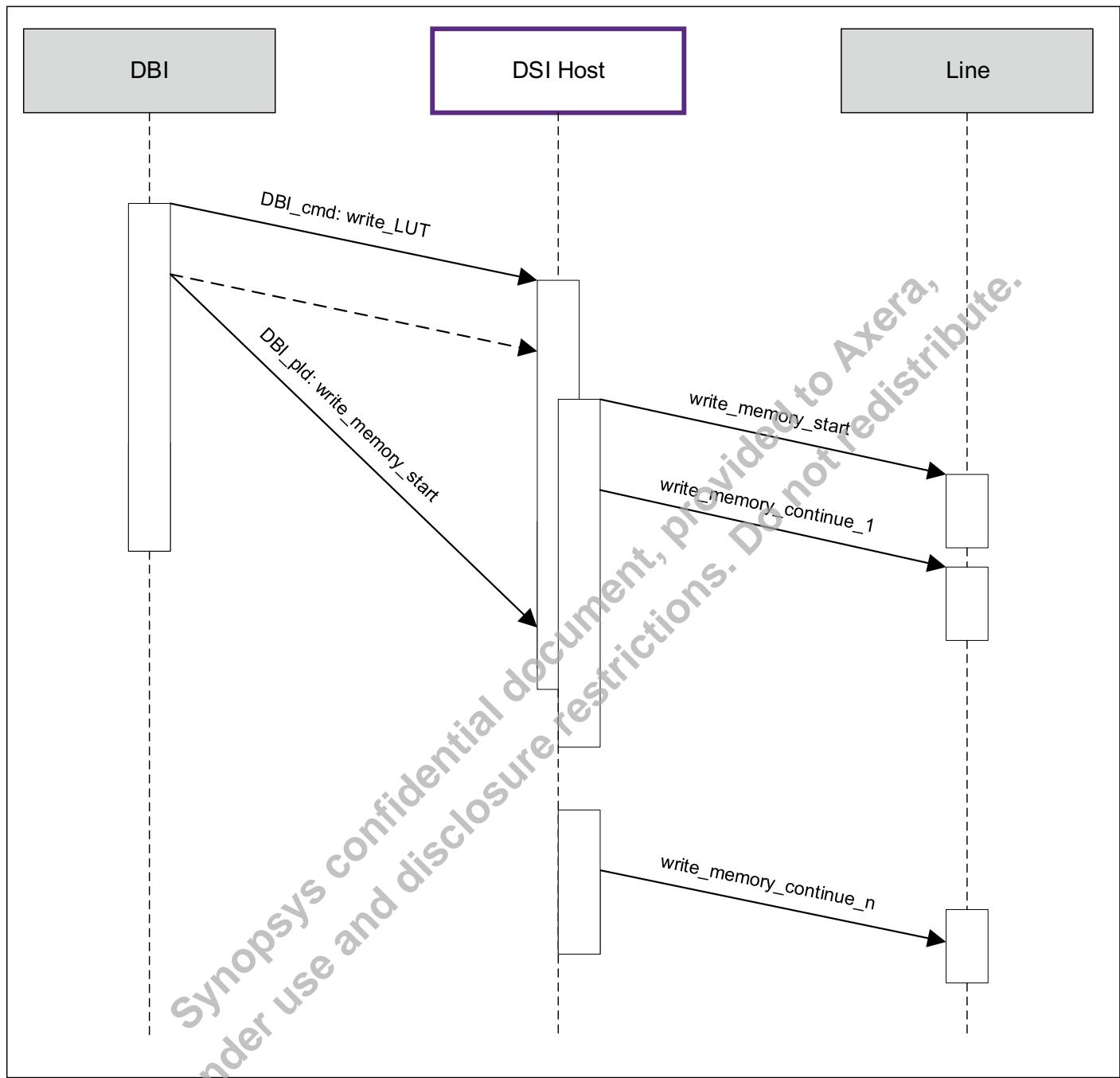
For more information, see the “Configuring the DBI Pixel Data” section in *DWC MIPI DSI Host Controller User Guide*.

2.5.5 Partitioning the Long write_memory_start Commands

The DWC_mipi_dsi_host controller divides a long DBI write_memory_start command into several packets of shorter length. This partitioning consists of one write_memory_start packet followed by several write_memory_continue packets. [Figure 2-20](#) illustrates this partitioning.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

Figure 2-20 Transmission of a Long write_memory_start Packet Partitioned by DSI Host Controller



This feature allows the DBI to issue multiple command packets to be transmitted to the DSI link. The controller binds the generated initial write_memory_start packet, and each write_memory_continue packet to a DSI packet. The value programmed in the `allowed_cmd_size` field of the `DBI_CMDSIZE` register, limits the packet size.

The `wr_cmd_size` field configures the full size of the DCS memory write command. For example, if `allowed_cmd_size = 200` and `wr_cmd_size = 900`, then the `DWC_mipi_dsi_host` controller sends five

memory write DCS commands. The first four commands are of 200 bytes length and the last command conveys the remaining 100 bytes. The values in these fields are measured in bytes.

[Table 2-3](#) shows the conversion of the number of pixels to bytes that must be used to correctly program the `wr_cmd_size` and `allowed_cmd_size` fields of the `DBI_CMDSIZE` register.

The number of bytes in a DCS memory write command must be an integer value. Therefore, the number of pixels, if using 12bpp, must be an even number. The 18-bpp pixel format does not require an even number of pixels, because according to the DCS Specification, the 18-bit format is zero padded to 24 bits that turns it into a format with three bytes per pixel. All other formats are divisible by eight and do not have any restrictions.

Table 2-3 Number of Pixels to Bytes Conversion for the DBI Pixels Formats

DBI	DBI_CFG Register	DBI_CMDSIZE Register	
Pixel Data Format	in_dbi_conf Field	wr_cmd_size Field	allowed_cmd_size Field
16-bit 12 bpp	0111	(3*number of pixels/2) +11	(3*number of pixels/2) +1 ¹
16-bit 16 bpp	1000	(2*number of pixels) + 1	(2*number of pixels) +1
16-bit 18 bpp, option 1	1001	(3*number of pixels) +1	(3*number of pixels) +1
16-bit 18 bpp, option 2	1010	(3*number of pixels) +1	(3*number of pixels) +1
16-bit 24 bpp, option 1	1011	(3*number of pixels) +1	(3*number of pixels) +1
16-bit 24 bpp, option 2	1100	(3*number of pixels) +1	(3*number of pixels) +1
16-bit 8 bpp	0110	(2*number of pixels) +1	number of pixels +1
9-bit 18 bpp	0101	(3*number of pixels) +1	(3*number of pixels) +1
8-bit 12 bpp	0001	(3*number of pixels/2) +11	(3*number of pixels/2) +11
8-bit 16 bpp	0010	(2*number of pixels) + 1	(2*number of pixels) +1
8-bit 18 bpp	0011	(3*number of pixels) +1	(3*number of pixels) +1
8-bit 24 bpp	0100	(3*number of pixels) +1	(3*number of pixels) +1
8-bit 8 bpp	0000	number of pixels +1	number of pixels +1

1. The number of pixels must be even so that the value programmed in the register field results in an integer number of bytes.

2.5.6 Handling the Low Transmission Speed

For slow transmission speed (for example, low-power transmission), the commands at the DBI might be generated faster than they are dispatched on the DSI link, causing the command FIFO to overflow. This happens because the DBI does not verify if the DBI command FIFO is full. This overflow causes an unrecoverable data loss, and requires a system soft reset.

To avoid this overflow situation, the system must monitor the state of the DBI command and the DBI payload FIFOs before writing more data. For instance, you can wait until both FIFOs are empty and then you can write an amount of data equal to the total size of the FIFO at once.

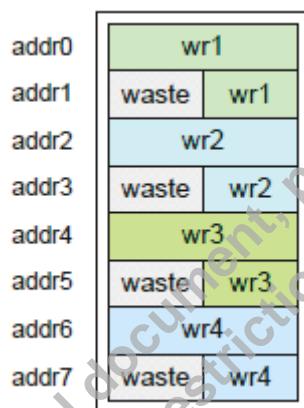
Alternatively, you can always keep the FIFO nearly full by monitoring the FIFO state until it is not full, and then write a single word of data. This solution is more resource consuming but it avoids FIFO starvation, making it possible to use FIFO sizes smaller than the amount of data of the longest packet to be written.

2.5.7 Dimensioning the Size of FIFOs for Automatic Partitioning

In the automatic partitioning controller configuration, the FIFOs size must be calculated considering that there may be some wastage due to the data organization.

For example, on setting the `allowed_cmd_size` packet, to partition data in parts of 6 bytes, each of these parts occupies one and a half positions of the RAM (the RAM data bus is 32 bits or 4 bytes). Data that belongs to different packets are not packed together in memory. Therefore, a wastage of 2 bytes of storage space per packet occurs in this case. [Figure 2-21](#) shows the schematic diagram of the memory contents for this example.

Figure 2-21 Example of Data Storage in DBI Payload FIFO



Automatic partition imposes minimum sizes for FIFOs. To send a total payload size of `pld_size` bytes with the maximum allowed size per packet set to `max_allowed` bytes, use the following expressions, to calculate the minimum size:



The operator denotes ceiling, or rounding up to the next integer.



$$\text{min_size_for_DBI_cmd_FIFO} = \left\lceil \frac{\text{pld_size}}{\text{max_allowed}} \right\rceil$$

$$\text{min_size_for_DBI_pld_FIFO} = \left\lceil \frac{\text{max_allowed}}{4} \right\rceil * \text{min_size_for_DBI_cmd_FIFO}$$

For example, consider the following values:

- `pld_size` = 24 bytes
- `max_allowed` = 6 bytes

Therefore,

$$\text{min_size_for_DBI_cmd_FIFO} = \left\lceil \frac{24}{6} \right\rceil = [4] = 4$$

$$\text{min_size_for_DBI_pld_FIFO} = \left\lceil \frac{6}{4} \right\rceil \times 4 = [1.5] \times 4 = 2 \times 4 = 8$$

[Figure 2-21](#) illustrates the DBI payload FIFO, which has exactly eight positions, which is the minimum number of positions required to send the whole payload data all at once.

2.5.8 Signals Related to DBI

For more information on signals related to DBI, see “Signal Descriptions” chapter.

2.5.9 Registers Related to DBI

- DBI_VCID
- DBI_CFG
- DBI_PARTITIONING_EN
- DBI_CMDSIZE

For more information on registers related to DBI, see “Register Descriptions” chapter.

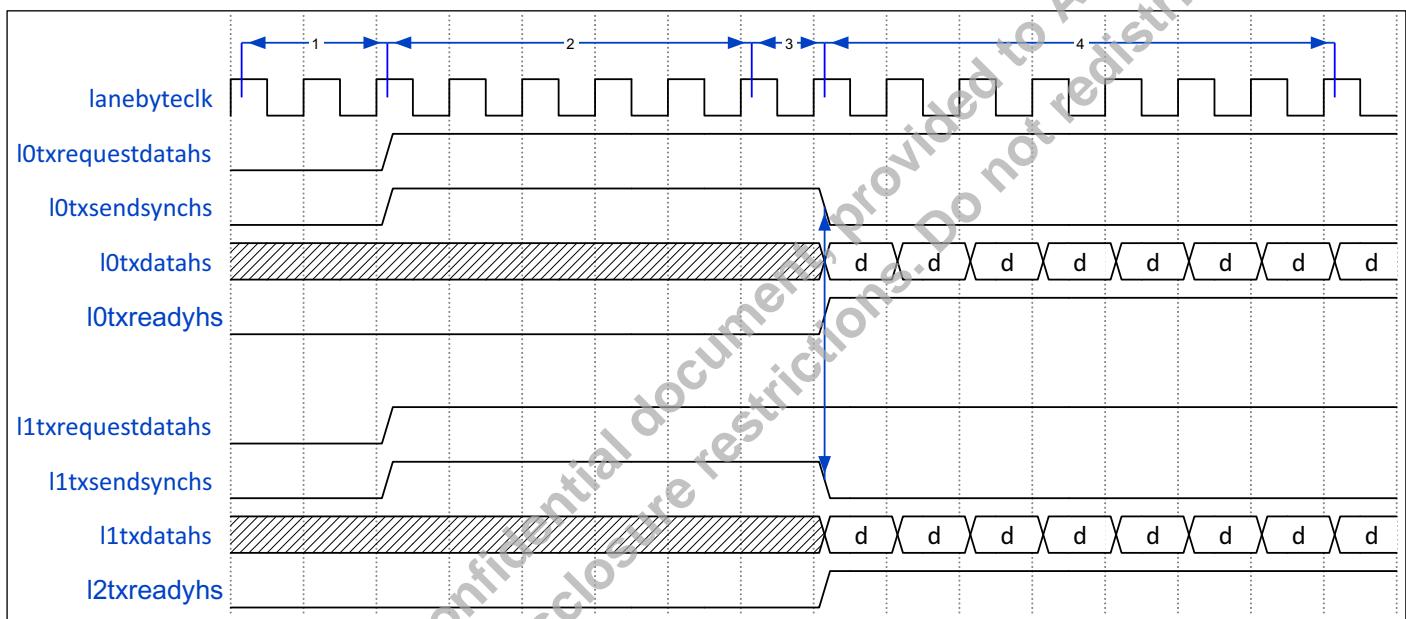
2.6 Continuous Clock Using C-PHY Mode

When using continuous clock in C-PHY mode, `I0txsendsynchs` signals must assert with `I0txrequestdatahs` signals, until assertion of `I0txreadyhs` signal.

For example,

1. As mentioned in the DSI Specification, lane 1 continuously sends SSS when there is no data to transfer
2. The `I0txrequestdatahs` and `I0txsendsynchs` signals assert, request lane 0 to enter into high-speed mode
3. PHY lane 0 is ready to receive High Speed data
4. The `DWC_mipi_dsi_host` controller sends valid data de-asserting both `I0txsendsynchs` and `I1txsendsynchs` signals

Figure 2-22 Example of Continuous Clock When Using Dual Lanes CPHY Mode



2.7 APB Slave Generic Interface

2.7.1 Overview of APB Slave Generic Interface

The APB Slave interface allows the transmission of generic information in Command mode, and follows the Synopsys proprietary register interface. Commands sent through this interface, need not be constrained to comply with the DCS Specification, and therefore, includes generic commands described in the DSI specification as manufacturer-specific.



Note The APB Slave Generic interface can operate concurrently with DPI, eDPI, or DBI.

2.7.2 Description of APB Slave Generic Interface

DesignWare Cores supports the transmission of write and read command mode packets, as described in the DSI specification. These packets are built using the APB register access.

Based on the operation, the functions of the GEN_PLD_DATA register are as follows:

- Writing into this register, sends the data as payload when sending a Command mode packet.
- Reading this register, returns the payload of a read back operation.

The GEN_HDR register contains the Command mode packet header type and header data. Writing into this register triggers the packet transmission implying that for a long Command mode packet, the packet's payload needs to be written in advance into the GEN_PLD_DATA register.

The valid packets available to be transmitted through the generic interface are as follows:

- Generic Write Short Packet 0 Parameters
- Generic Write Short Packet 1 Parameters
- Generic Write Short Packet 2 Parameters
- Generic Read Short Packet 0 Parameters
- Generic Read Short Packet 1 Parameters
- Generic Read Short Packet 2 Parameters
- Maximum Read Packet Configuration
- Generic Long Write Packet
- DCS Write Short Packet 0 Parameters
- DCS Write Short Packet 1 Parameters
- DCS Read Short Packet 0 Parameters
- DCS Write Long Packet

A set of bits in the CMD_PKT_STATUS register report the status of the FIFOs associated with APB interface support.

The controller transports the generic interface packets using one of the DSI transmission modes: Video mode or Command mode. If neither of these modes is selected, the controller does not transmit the packets through the link, and the related FIFOs eventually overflows.



The DPI and DBI does not support vendor-specific commands, which requires display initialization. Selecting the eDPI, automatically selects the generic interface.

2.7.3 Packet Transmission Using the Generic Interface

The packet transfer through the APB bus is based on the following conditions:

- The APB protocol defines that the write and read procedure takes two clock cycles each to be executed. This means that the maximum input data rate through the APB interface is always half the speed of the APB clock.
- The data input bus has a maximum width of 32 bits. This allows for a relation to be defined between the input APB clock frequency and the maximum bit rate achievable by the APB interface.
- When using only APB, the DSI link pixel bit rate is:
$$\text{DSI link pixel bit rate} = (\text{APB clock frequency}) * 16 \text{ Mbps}$$
- When using only the APB interface, the theoretical DSI link maximum bit rate is:
$$\text{DSI link maximum bit rate} = \text{APB clock frequency (in MHz)} * 32 / 2 \text{ Mbps}$$
In the above equation, the number 32 represents the APB data bus width. The division by two is because each APB write procedure takes two clock cycles to be executed.
- The available bandwidth depends on the APB clock frequency, and increases with the clock frequency.

To drive the APB interface to achieve high bandwidth Command mode traffic transported by the DSI link, the DWC_mipi_dsi_host controller must operate only in the Command mode, and the APB interface must be the only data source currently in use. Thus, the APB interface has the entire bandwidth of the DSI link and does not share it with any another input interface source.

The memory write commands require maximum throughput from the APB interface, because they contain the most amount of data conveyed by the DSI link. While writing the packet information, first write the payload of a given packet into the payload FIFO using the GEN_PLD_DATA register. For the payload data of the command parameters, place the first byte to be transmitted in the least significant byte position of the APB data bus. For more information, see GEN_PLD_DATA register in the “Register Descriptions”.

After writing the payload, write the packet header into the command FIFO, so that it is correctly stored inside the Command FIFO. For more information about the packet header organization on the 32-bit APB data bus, see GEN_HDR register in the “Register Descriptions”.

The payload data for a memory write command contains pixel information and it must follow the pixel to byte conversion organization referred in the DCS Specification's Annex A. [Figure 2-23](#) to [Figure 2-27](#) shows how the pixel data must be organized in the APB data write bus. The memory write commands are conveyed in DCS long packets, which are encapsulated in a DSI packet. The DSI specifies that the DCS command must be present in the first payload byte of the packet. This is also included in the diagrams. In [Figure 2-23](#) to [Figure 2-27](#), the Write Memory Command can be replaced by the DCS command Write Memory Start and Write Memory Continue.

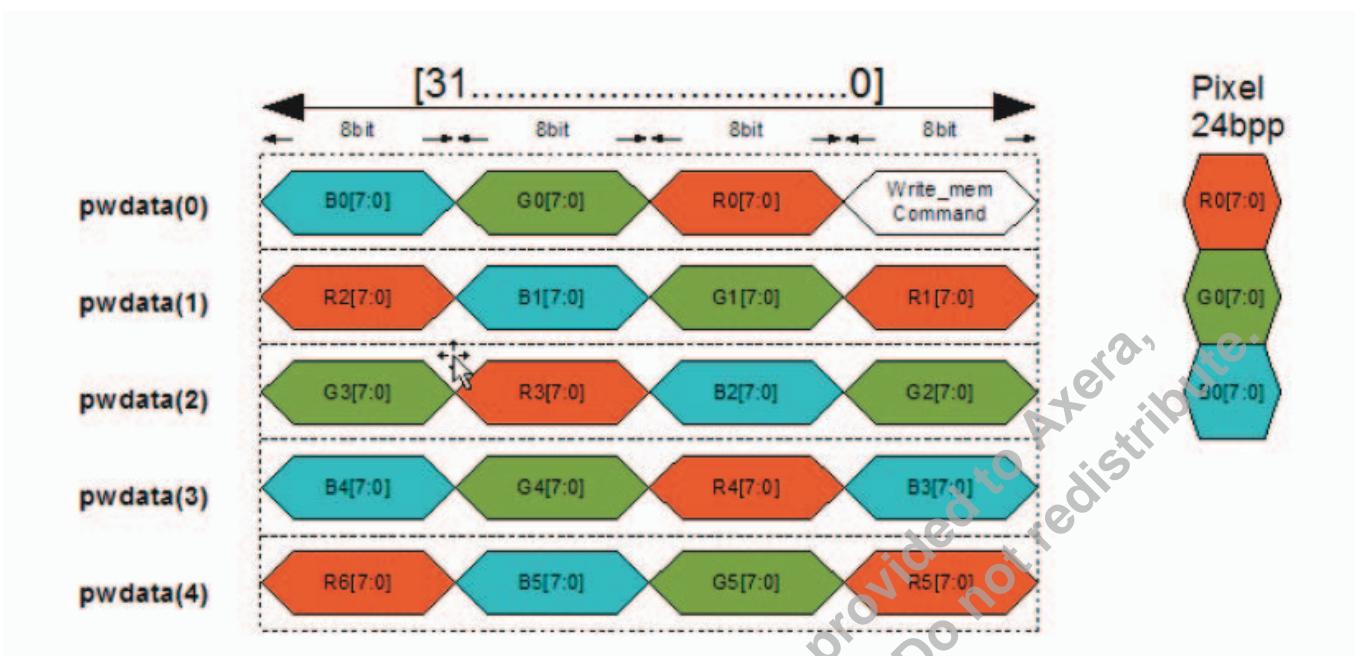
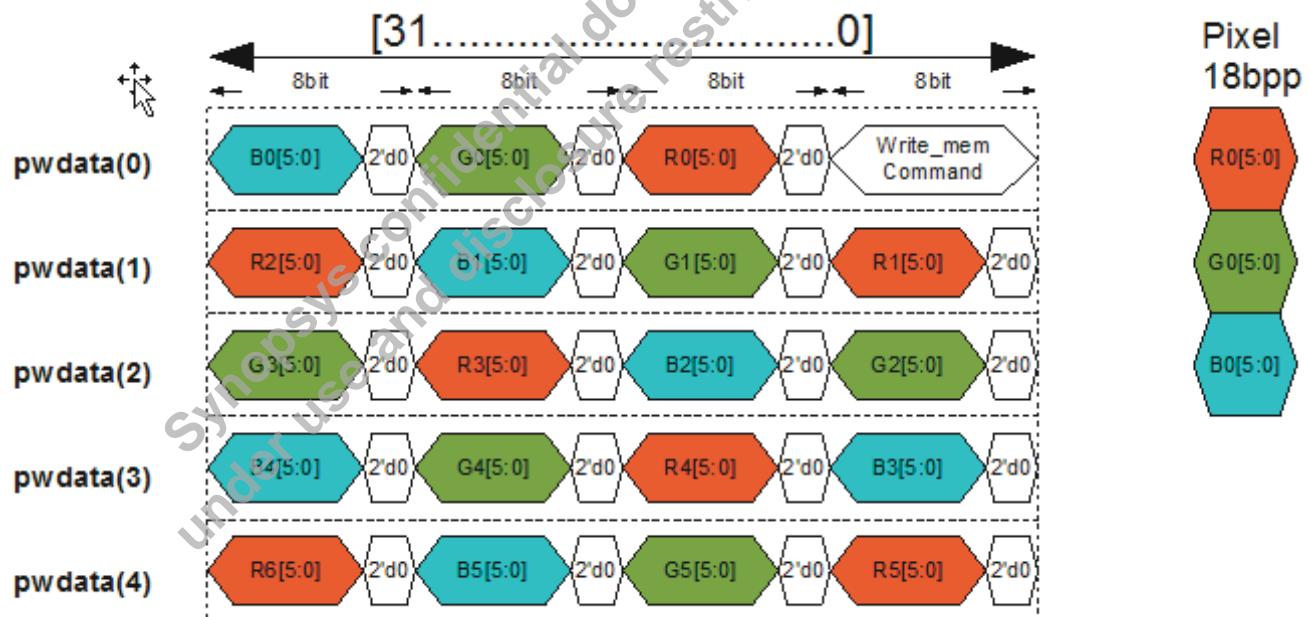
Figure 2-23 24 bpp APB Pixel to Byte Organization**Figure 2-24 18 bpp APB Pixel to Byte Organization**

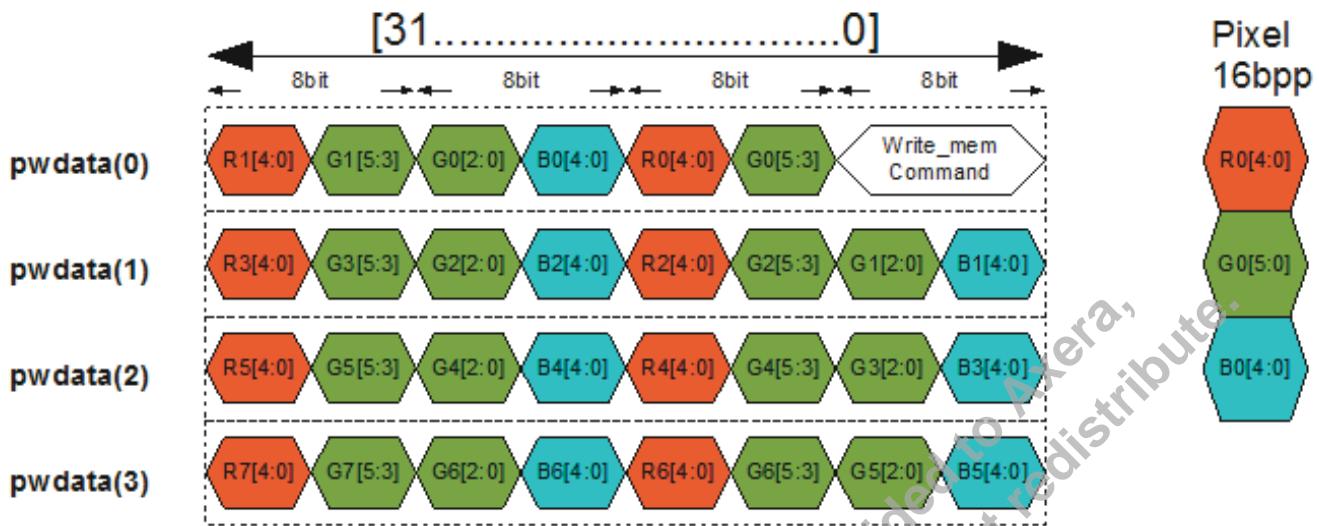
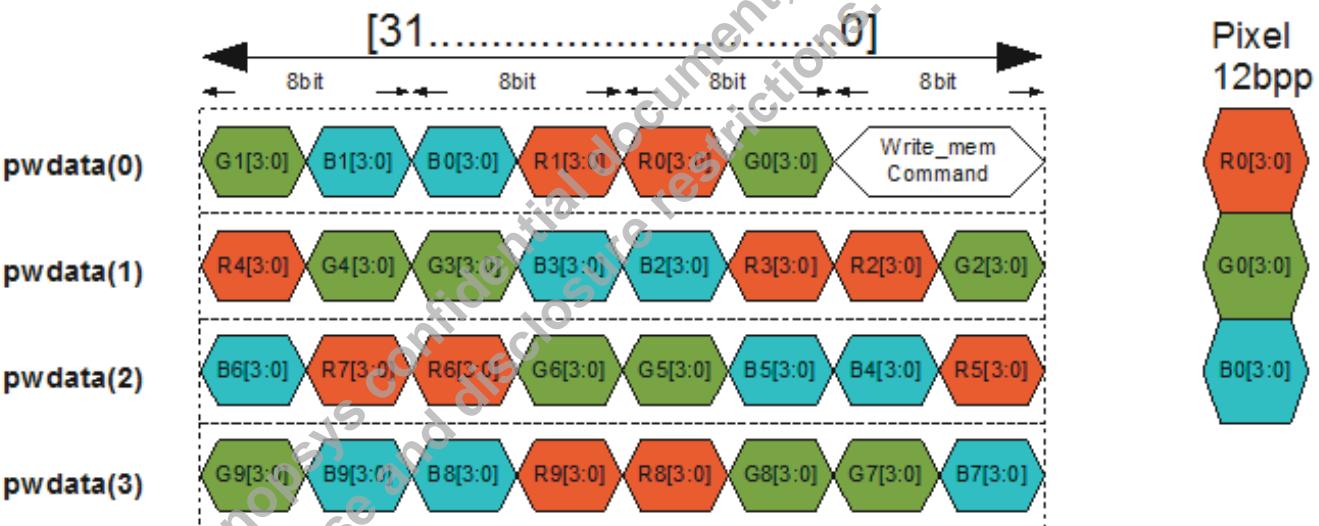
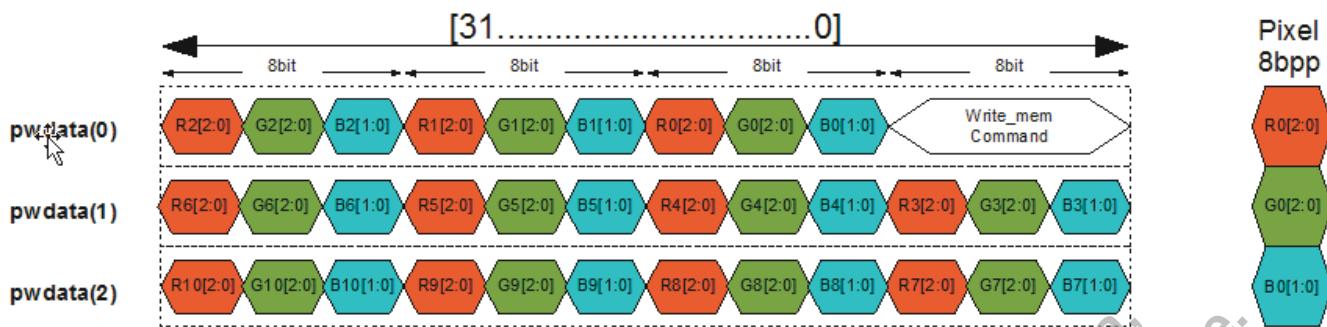
Figure 2-25 16 bpp APB Pixel to Byte Organization**Figure 2-26 12 bpp APB Pixel to Byte Organization**

Figure 2-27 8 bpp APB Pixel to Byte Organization

2.7.4 Signals Related to APB Slave Generic Interface

For more information about the signals related to the APB Slave Generic Interface, see “Signal Descriptions” chapter.

2.7.5 Registers Related to APB Slave Generic Interface

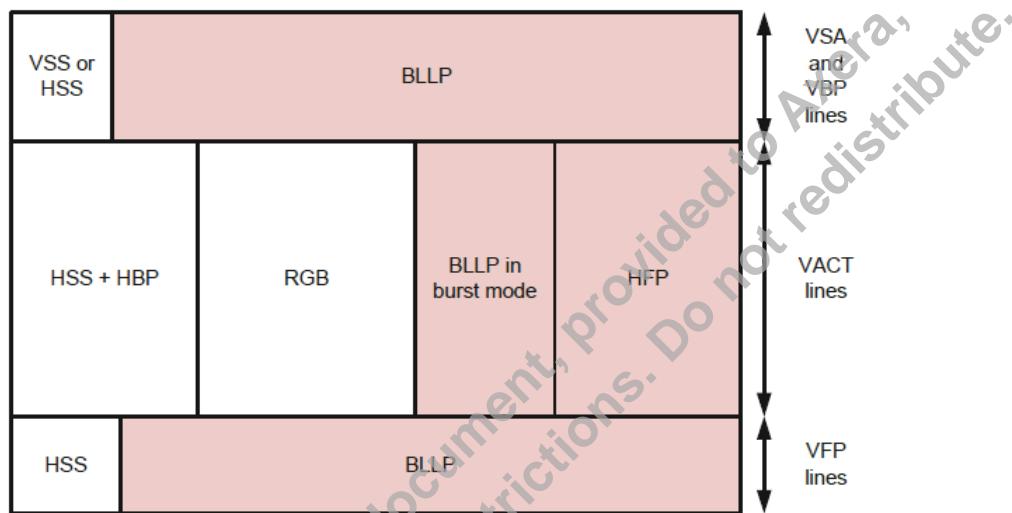
For more information about the registers related to the APB Slave Generic Interface, see “Register Descriptions” chapter.

2.8 Transmission of Commands

2.8.1 Transmission of Commands in Video Mode

The DWC_mipi_dsi_host controller supports the transmission of commands, both in high-speed and low-power, while in Video mode. The DWC_mipi_dsi_host controller uses Blanking or Low-power (BLLP) periods to transmit commands inserted through the APB Generic interface. Those periods correspond to the shaded areas of [Figure 2-28](#).

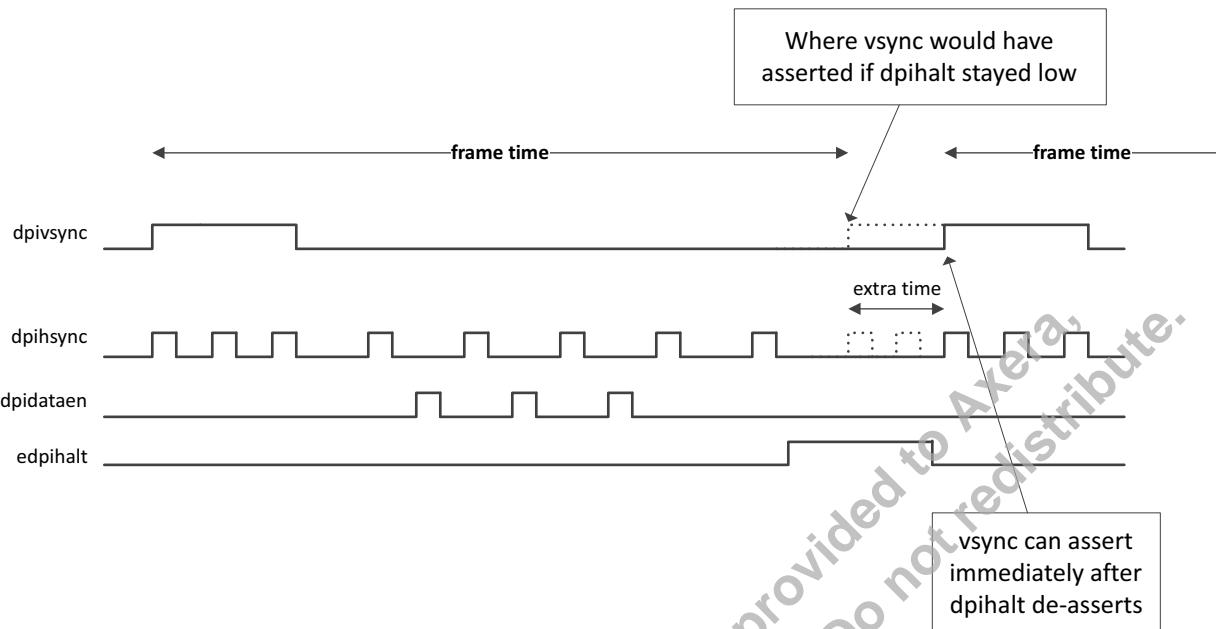
Figure 2-28 Command Transmission Periods within the Image Area



Commands are transmitted in the blanking periods after the following packets or states:

- Vertical Sync Start (VSS) packets, if the Video Sync pulses are not enabled
- Horizontal Sync End (HSE) packets, in the VSA, VBP, and VFP regions
- Horizontal Sync Start (HSS) packets, if the Video Sync pulses are not enabled in the VSA, VBP, and VFP regions
- Horizontal Active (HACT) state

Besides the areas corresponding to BLLP, it is possible to send large commands during the last line of a frame. But this violates the line time for the Video mode and sets the `edpihalt` signal to request the DPI video timing signals to remain inactive. If a command does not fit into any BLLP area, it is postponed to the last line, causing the violation of the line time for the Video mode, as illustrated in [Figure 2-29](#).

Figure 2-29 Transmission of Commands on the Last Line of a Frame

The controller transmits only one command per line, even in the case of the last line of a frame.

In low-power, the controller sends only one command per line. In high-speed, the DWC_mipi_dsi_host controller can send more than one command, as many as it determines to fit in the available time.

The DWC_mipi_dsi_host controller avoids sending commands in the last line because the last line may be shorter than the others. For instance, the line time (t_L) can be half a cycle longer than the t_L on the DPI, that is, each line in the frame takes half a cycle from time for the last line. This results in the last line being $(\frac{1}{2} \text{ cycle}) \times (\text{number of lines} - 1)$ shorter than t_L .

On transmitting a command, the **edpihalt** signal asserts in the last line. The **dpivsync_edpiwms** signal asserts immediately after **edpihalt** de-asserts.

The **dpicolorm** and **dpishutdn** input signals also triggers the sending of command packets. The commands are DSI data types Color Mode On, Color Mode Off, Shut Down Peripheral, and Turn on Peripheral. These commands must not be sent in the VACT region. On setting the **lp_cmd_en** bit of the **VID_MODE_CFG** register to 1, these commands are sent in low-power mode. In low-power mode, the **outvact_lpcmd_time** field of the **DPI_LP_CMD_TIM** register determines if these commands can be transmitted. It is assumed that **outvact_lpcmd_time** is greater than or equal to four bytes (number of bytes in a short packet), because the DWC_mipi_dsi_host controller does not transmit these commands on the last line.

If the **frame_bta_ack_en** field of the **VID_MODE_CFG** register is set, then the DWC_mipi_dsi_host controller generates a BTA after the last line of a frame. This may coincide with a write command or a read command. In either case, the controller asserts the **edpihalt** signal, until an acknowledge is received (control of the DSI bus is returned to the host).

2.8.2 Transmission of Commands in Low-Power Mode

The DWC_mipi_dsi_host controller can be configured to send the low-power commands during the high-speed video mode transmission.

To enable this feature, set the lp_cmd_en bit of the VID_MODE_CFG register to 1.

In this case, it is necessary to calculate the time available in bytes, to transmit a command in low-power mode to Horizontal Front Porch (HFP), Vertical Sync Active (VSA), Vertical Back Porch (VBP), and Vertical Front Porch (VFP) regions.

Bits 8 to 13 of the VID_MODE_CFG register indicates if the DWC_mipi_dsi_host controller can go to low-power when in idle. If the lp_cmd_en bit is set (1'b1) and non-video packets are in queue, the DWC_mipi_dsi_host controller ignores the low-power configuration and transmits low-power commands, even if it is not allowed to enter low-power in a specific region. After the low-power commands transmission, the DWC_mipi_dsi_host controller remains in low-power until a sync event occurs.

For example, consider VFP selected as high-speed region ($lp_{vfp_en} = 1'b0$) with lp_{cmd_en} set as a command to transmit in low-power in the VPF region. Then the controller transmits this command in low-power, and the line stays in low-power until a new HSS arrives.

2.8.2.1 Calculating the Time to Transmit Commands in Low-Power Mode in the VSA, VBP, and VFP Regions

The outvact_lpcmd_time field of the DPI_LP_CMD_TIM register indicates the time available (in bytes) to transmit a command in low-power mode (based on the escape clock) on a line during the VSA, VBP, and the VFP regions.

Calculation of outvact_lpcmd_time depends on the Video mode that you use. [Figure 2-30](#) illustrates the timing intervals for the Video mode in non-burst with sync pulses and [Figure 2-31](#) illustrates the timing intervals for the Video mode in burst and non-burst with sync events.

Figure 2-30 outvact_lpcmd_time for Non-Burst with Sync Pulses

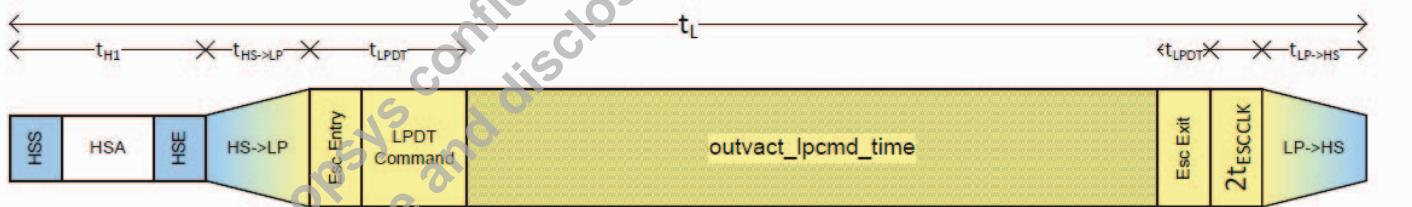
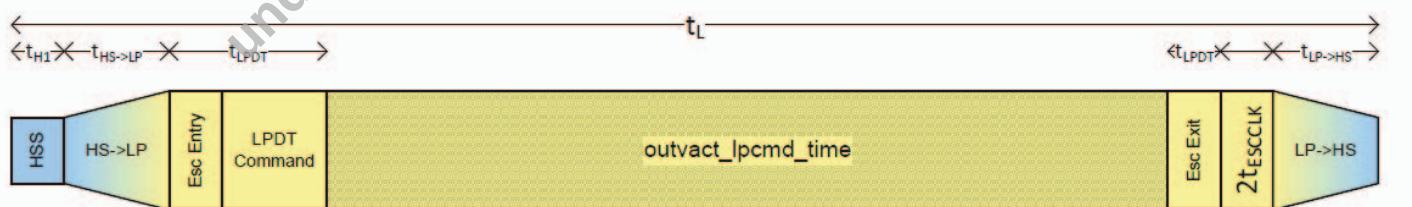


Figure 2-31 outvact_lpcmd_time for Burst or Non-Burst with Sync Events



This time calculation, is as follows:

$$\text{outvact_lpcmd_time} = (\text{tL} - (\text{tH1} + \text{tHS-} \rightarrow \text{LP} + \text{tLP-} \rightarrow \text{HS} + \text{tLPDT} + 2\text{tESCCLK})) / (2 \times 8 \times \text{tESCCLK})$$

Where,

- tL = Line time
- tH1 = Time of the HSA pulse for sync pulses mode ([Figure 2-30](#)) or the time to send the HSS packet, including EoTp ([Figure 2-31](#))
- $\text{tHS-} \rightarrow \text{LP}$ = Time to enter the low-power mode
- $\text{tLP-} \rightarrow \text{HS}$ = Time to leave the low-power mode
- tLPDT = PHY timing related with Escape Mode Entry, LPDT Command, and Escape Exit. According to the PHY specification, this value is always 11 bits in low-power (or 22 TX Escape clock cycles).
- tESCCLK = Escape clock period as programmed in the `tx_esc_clk_division` field of the `CLKMGR_CFG` register
- 2tESCCLK = Delay imposed by the controller implementation

In the above equation, division by eight is done to convert the time available to bytes. Division by two is done because one bit is transmitted once in every two escape clock cycles. The `outvact_lpcmd_time` field can be compared directly with the size of the command to be transmitted to determine if there is enough time to transmit the command. This field limits the maximum size of a command that can be transmitted in low-power mode to 255 bytes. Program this register to a value greater than or equal to 4 bytes for the transmission of the DCTRL commands, such as shutdown and color in low-power mode.

For example, consider a frame with 12.4 μ s per line and assume an escape clock frequency of 20 MHz and a lane bit rate of 800 Mbits. In this case, it is possible to send 124 bits in escape mode (that is, 124 bit = 12.4 μ s * 20 MHz/2). Still, you need to take into consideration the PHY protocol and PHY timings. The assumptions are as follows:

- Lane byte clock period is 10 ns (800 Mbits per Lane)
- Escape clock period is 50 ns (`CLKMGR_CFG`: `tx_esc_clk_division` = 5)
- Video is transmitted in non-Burst mode with sync pulses bounded by HSS and HSE packets
- DSI is configured for two Lanes
- PHY takes 180 ns to transit from low-power to high-speed modes (`PHY_TMR_CFG`: `phy_lp2hs_time` = 18)
- PHY takes 200 ns to transit from high-speed to low-power modes (`PHY_TMR_CFG`: `phy_hs2lp_time` = 20)
- $\text{tHSA} = 420$ ns

In this example, a 13-byte command can be transmitted as follows:

$$\text{outvact_lpcmd_time} = (12.4 \mu\text{s} - (420 \text{ ns} + 180 \text{ ns} + 200 \text{ ns} + (22 \times 50 \text{ ns} + 2 \times 50 \text{ ns}))) / (2 \times 8 \times 50 \text{ ns}) = 13 \text{ bytes}$$

2.8.2.2 Calculating the Time to Transmit the Commands in Low-Power Mode in the HFP Region

The `invact_lpcmd_time` field of the `DPI_LP_CMD_TIM` register indicates the time available (in bytes) to transmit a command in low-power mode (based on the escape clock) in the Vertical Active (VACT) region. To calculate the value of `invact_lpcmd_time`, consider the video mode that you use. [Figure 2-32](#) shows

the timing intervals for video mode in Non-Burst with sync pulses. [Figure 2-33](#) shows the timing intervals for video mode in non-Burst with sync events. [Figure 2-34](#) shows the Burst video mode.

Figure 2-32 invact_lpcmd_time for Non-Burst with Sync Pulses

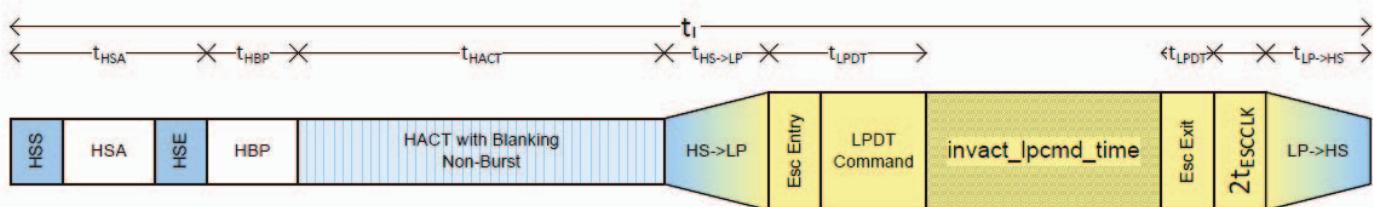


Figure 2-33 invact_lpcmd_time for Non-Burst with Sync Events

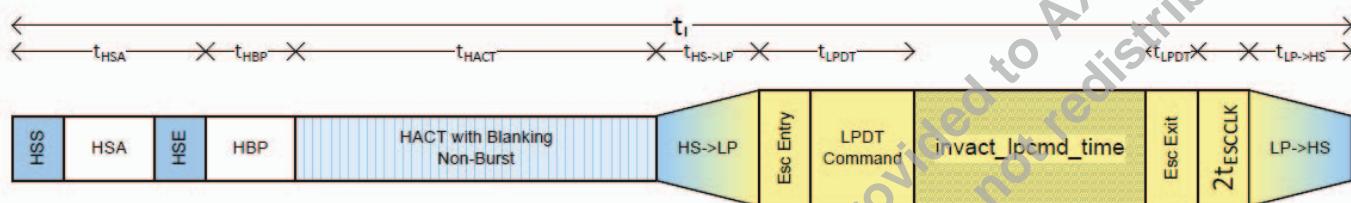
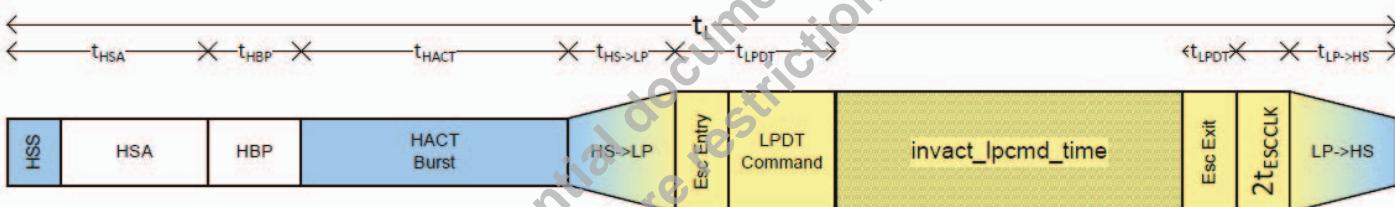


Figure 2-34 invact_lpcmd_time for Burst Mode



This time calculation, is as follows:

$$\text{invact_lpcmd_time} = (tL - (tHSA + tHBP + tHACT + tHS->LP + tLP->HS + tLPDT + 2tESCCLK)) / (2 \times tESCCLK)$$

Where

- tL = Line time
- $tHSA$ = Time of the HSA pulse (VID_HSA_TIME)
- $tHBP$ = Time of Horizontal back porch (VID_HBP_TIME)
- $tHACT$ = Time of Video active. For Burst mode, the Video active is time compressed and is calculated as follows:

$$tHACT = \text{vid_pkt_size} * \text{Bytes_per_Pixel} / \text{Number_Lanes} * tLane_byte_clk$$
- $tESCCLK$ = escape clock period as programmed in tx_esc_clk_division field of the CLKMGR_CFG register

The invact_lpcmd_time field can be compared directly with the size of the command to be transmitted to determine if there is time to transmit the command.

For example, consider a frame with 16.4 μ s per line and assume an escape clock frequency of 20 MHz and a Lane bit rate of 800 Mbits. In this case, it is possible to send 420 bits in escape mode (that is, 164 bits = 16.4 μ s * 20 MHz/2). Because this is the Vertical Active region of the frame, consider the HSA, HBP, and HACT timings apart from the D-PHY protocol and PHY timings. The assumptions, are as follows:

- PHY: D-PHY
- Number of active lanes is 4
- Lane byte clock period (lanebyteclkperiod) is 10 ns (800 Mbits per Lane)
- Escape clock period is 50 ns (CLKMGR_CFG: tx_esc_clk_division = 5)
- PHY takes 180 ns to transit from low-power to high-speed modes (PHY_TMR_CFG: phy_lp2hs_time = 18)
- PHY takes 200 ns to transit from high-speed to low-power modes (PHY_TMR_CFG: phy_hs2lp_time = 20)
- tHSA = 420 ns
- tHBP = 800 ns
- tHACT = 12800 ns to send 1280 pixel at 24 bpp
- Video is transmitted in non-Burst mode
- DWC_mipi_dsi_host controller is configured for four lanes

In this example, consider sending the video in Non-burst mode. The invact_lpcmd_time is calculated as:

$$\text{invact_lpcmd_time} = (16.4 \text{ us} - (420 \text{ ns} + 800 \text{ ns} + 12.8 \text{ us} + 180 \text{ ns} + 200 \text{ ns} + (22 \times 50 \text{ ns} + 2 \times 50 \text{ ns})) / (2 \times 8 \times 50 \text{ ns}) = 1 \text{ byte}$$

Therefore, it is possible to transmit only one byte in this period. A short packet (for example, generic short write) requires a minimum of four bytes. Therefore, in this example, commands are not sent in the VACT region.

If Burst mode is enabled, more time is available to transmit the commands in the VACT region, because HACT is time compressed.

$$\text{invact_lpcmd_time} = (16.4 \text{ us} - (420 \text{ ns} + 800 \text{ ns} + (1280 \times 3 / 4 \times 10 \text{ ns}) + 180 \text{ ns} + 200 \text{ ns} + (22 \times 50 \text{ ns} + 2 \times 50 \text{ ns})) / (2 \times 8 \times 50 \text{ ns}) = 5 \text{ bytes}$$

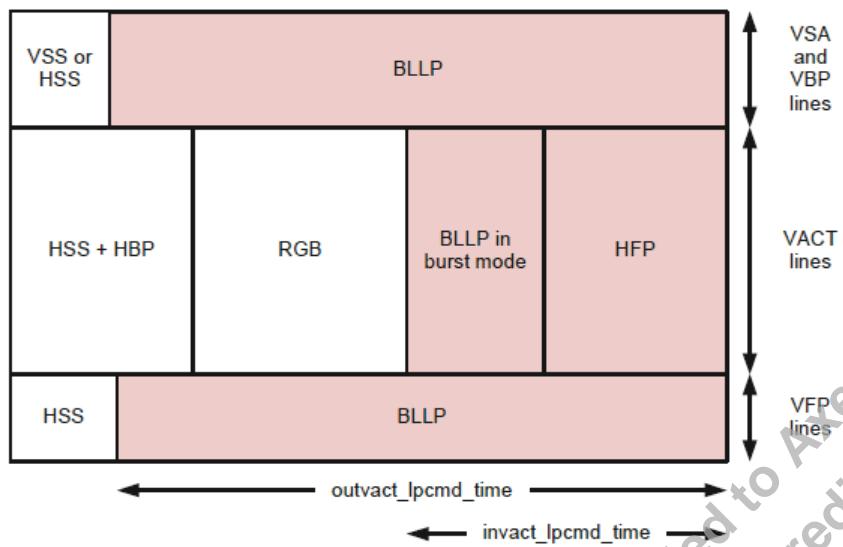
Thus for Burst mode, the invact_lpcmd_time is 5 bytes and you can effectively send a 4-byte short packet.

2.8.2.3 Transmission of Commands in Different Periods

The outvact_lpcmd_time and invact_lpcmd_time fields allow a simple comparison to determine if a command can be transmitted in any of the BLLP periods.

[Figure 2-35](#) illustrates the meaning of invact_lpcmd_time and outvact_lpcmd_time, matching them with the shaded areas and the VACT region.

Figure 2-35 Location of outvact_lpcmd_time and invact_lpcmd_time in the Image Area



2.8.3 Transmission of Commands in High-Speed Mode

The controller sends the commands in high-speed in Video Mode, if the `lp_cmd_en` bit of the `VID_MODE_CFG` register is 0. In this case, the `DWC_mipi_dsi_host` controller, automatically determines the area where each command can be sent and requires no programming or calculation.

2.8.4 Read Command Transmission

The `max_rd_time` field of the `PHY_TMR_RD_CFG` register configures the maximum amount of time required to perform a read command in lane byte clock cycles. It is calculated as follows:

$\text{max_rd_time} = \text{Time to transmit the read command in low-power mode} + \text{Time to enter and leave low-power mode} + \text{Time to return the read data packet from the peripheral device}$

The time to return the read data packet from the peripheral depends on the number of bytes read and the escape clock frequency of the peripheral, not the escape clock of the host. In both the high-speed and low-power mode, the `max_rd_time` field, determines if there is time to complete a read command in a BLLP period.

In high-speed mode (`lp_cmd_en = 0`), `max_rd_time` is calculated as follows:

$$\text{max_rd_time} = (\text{tHS-} \rightarrow \text{LP} + \text{tLP-} \rightarrow \text{HS} + \text{tread} + 2 \times \text{tBTA}) / \text{lanebyteclkperiod}$$

In low-power mode (`lp_cmd_en = 1`), `max_rd_time` is calculated as follows:

$$\text{max_rd_time} = (\text{tHS-} \rightarrow \text{LP} + \text{tLP-} \rightarrow \text{HS} + \text{tLPDT} + \text{tlprd} + \text{tread} + 2 \times \text{tBTA}) / \text{lanebyteclkperiod}$$

Where,

- $\text{tHS-} \rightarrow \text{LP}$ = Time to enter the low-power mode
- $\text{tLP-} \rightarrow \text{HS}$ = Time to leave the low-power mode

- tLPDT = PHY timing related to Escape mode entry, LPDT command, and Escape mode exit. According to the PHY specification, this value is always 11 bits in low-power (or 22 TX escape clock cycles).
- tlprd = Read command time in low-power mode (64 * TX esc clock)
- tread = Time to return the read data packet from the peripheral
- tBTA = time to perform a bus turnaround (PHY dependent)

It is recommended to keep the maximum number of bytes read from the peripheral to a minimum, to have sufficient time available to issue the read commands in a line time.

To avoid sending read commands on the last line of a frame, ensure the following:

$(\text{max_rd_time} \times \text{Lane byte clock period}) < (\text{outvact_lpcomd_time} \times 16 \times \text{escape clock period of the host})$

If it is necessary to read a large number of parameters (> 16), increase the `max_rd_time` while the read command is being executed. On completion of read operation, decrease the `max_rd_time` value.

Issuing a read command on the last line of a frame, asserts the `edpihalt` signal, and remains asserted until the read command is in progress, during which the video transmission must be stopped.

2.8.5 Clock Lane in Low-Power Mode (D-PHY)

To reduce D-PHY power consumption, the `DWC_mipi_dsi_host` controller, when not transmitting in the high-speed mode, allows clock lane transition to the low-power mode. The controller automatically handles the transition of the clock lane from high-speed (clock lane active sending clock) to low-power state without direct intervention by the software. Configuring the `phy_txrequestclkhs` and the `auto_clklane_ctrl` bits of the `LPCLK_CTRL` register, to enable this feature.

In the Command mode, it is possible for the `DWC_mipi_dsi_host` controller, to place the clock lane in the low-power mode, when it does not have any high-speed packets to transmit.

In the Video mode (DPI), the `DWC_mipi_dsi_host` controller uses its internal video and PHY timing configurations to determine if there is time available for the clock line to enter low-power mode, while not compromising video data transmission of pixel data and sync events.

Along with a correct configuration of the Video mode (for more information about DPI, see “[DPI](#)” on page [41](#)), the `DWC_mipi_dsi_host` controller needs to know the time required by the clock lane to go from high-speed to low-power and from low-power to high-speed. The values required can be obtained from the D-PHY documentation. If you use Synopsys D-PHY, see DesignWare Cores MIPI Bidir 4L D-PHY Databook.

Program the `PHY_TMR_LPCLK_CFG` register with the following values:

- `phy_clkhs2lp` = Time from high-speed to low-power in clock lane / Byte clock period in high-speed (`lanebybyteclk`)
- `phy_clklp2hs` = Time from low-power to high-speed in clock lane / Byte clock period in high-speed (`lanebybyteclk`)

Based on the programmed values, the `DWC_mipi_dsi_host` controller, calculates if there is enough time for the clock lane to enter the low-power mode, during inactive regions of the video frame.

There is an exception where the clock lane is activated even when there is no high-speed packet to be transmitted. If a command is not allowed to be transmitted in any of the available blanking periods, it is transmitted during the last line of the frame. This situation causes `edpihalt` signal assertion, indicating to the video engine to halt the video generation. Upon `edpihalt` signal assertion, the controller requests the

clock lane to start transmission of high-speed clock. This is to avoid delay in the transmission of the next frame vertical synchronism event, by the clock lane low-power to high-speed transition delay. For more information, see [Figure 2-29](#) on page [77](#).

The DWC_mipi_dsi_host controller decides the best approach, for power saving, from the following possible scenarios:

- There is no sufficient time to go to the low-power mode. Therefore, blanking period is added as shown in [Figure 2-36](#).
- There is sufficient time for the data lanes to go to the low-power mode but not enough time for the clock lane to enter the low-power mode. [Figure 2-37](#) illustrates this.
- There is sufficient time for both data lanes and the clock lane to go to the low-power mode. [Figure 2-38](#) illustrates this.

Figure 2-36 Clock Lane and Data Lanes in High-Speed

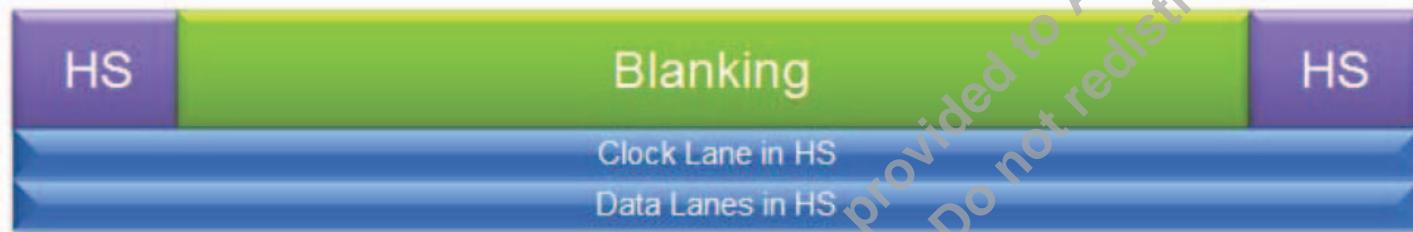


Figure 2-37 Clock Lane in High-Speed and Data Lanes in Low-Power



Figure 2-38 Clock Lane and Data Lanes in Low-Power



2.9 Virtual Channels

2.9.1 Overview of Virtual Channels

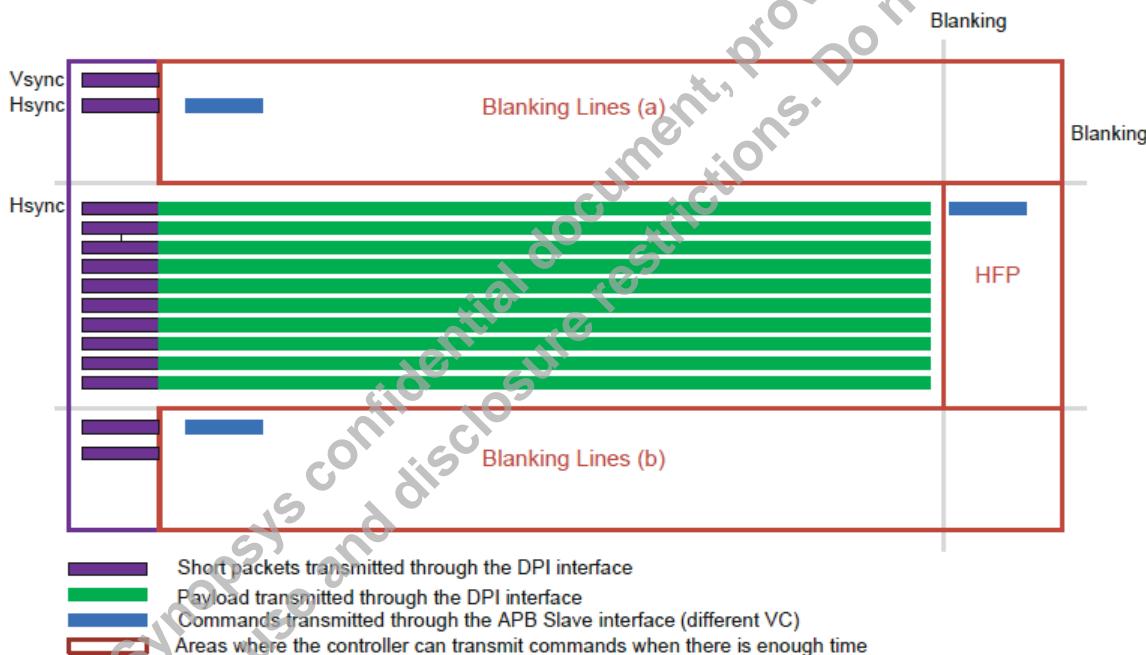
The DWC_mipi_dsi_host controller supports a specific virtual channel for each interface, enabling the system to address multiple displays simultaneously.

On configuring the DPI for a particular Virtual Channel, it is possible to use the APB Slave Generic interface to issue commands, while the video stream is being transmitted. By doing this, it is possible to send the commands through the ongoing video stream, addressing different virtual channels and thus enable the interface with multiple displays.

During the Video mode, the video stream transmission has the maximum priority. Therefore, the controller transmits sideband packets such as the ones from the generic interface, only when there is time available within the video stream transmission.

The DWC_mipi_dsi_host controller identifies the available time periods and uses them to transport the generic interface packets. [Figure 2-39](#) illustrates where the DWC_mipi_dsi_host controller inserts the packets from the APB Generic interface within the video stream transmitted by the DPI.

Figure 2-39 Command Transmission by the Generic Interface



On configuring the DBI for a particular virtual channel, it is possible to use the APB Slave Generic Interface to issue more commands while in Command mode transmission. This is also possible with multiple displays (and subsequently with multiple virtual channels). The Command mode does not have particular timing requirements as opposed to Video mode, the priority is given to the DBI generated packets and only then considers the Generic interface packets. This means that as long as there are packets from the DBI in queue for transmission, the Generic interface packets have to wait.

It is also possible to address the multiple displays with only the generic interface using different Virtual Channels. The configuration does not restrict the generic interface to any particular virtual channel, and

therefore, it is possible to issue packets with different virtual channels. This enables the interface to time multiplex the packets to be provided to the displays with different virtual channels.

2.9.2 Registers Related to Virtual Channels

Use the following configuration registers to select the virtual channel ID associated with transmissions over the DPI, DBI, and APB Slave Generic interfaces:

- **DPI_VCID:** The dpi_vc_id field of **DPI_VCID** register, configures the DPI virtual channel ID indexed to the Video mode packets.
- **DBI_VCID:** The dbi_vc_id field of **DBI_VCID** register, configures the DBI virtual channel ID indexed to the DCS packets.
- **GEN_HDR:** This register configures the packet header (which includes the virtual channel ID to be used) for transmissions using APB Slave Generic interface.

2.10 Video Mode Pattern Generator

2.10.1 Overview of Video Mode Pattern Generator

The Video Mode Pattern Generator allows the transmission of horizontal or vertical color bar and PHY BER testing pattern without any stimuli.

[Table 2-4](#) lists the frame requirements that must be defined in the video registers.

Table 2-4 Frame Requirement Configuration Registers

Register Name	Description
VID_MODE_CFG	Video mode configuration
VID_PKT_SIZE	Video packet size
VID_NUM_CHUNKS	Number of chunks
VID_NULL_SIZE	Null packet size
VID_HSA_TIME	Horizontal sync active time
VID_HBP_TIME	Horizontal back porch time
VID_HLINE_TIME	Line time
VID_VSA_LINES	Vertical sync active period
VID_VBP_LINES	Vertical back porch period
VID_VFP_LINES	Vertical front porch period
VID_VACTIVE_LINES	Vertical resolution

2.10.2 Registers Related to Video Mode Pattern Generator

For more information, see the following registers related to video mode pattern generator:

- VID_MODE_CFG
- VID_PKT_SIZE
- VID_NUM_CHUNKS
- VID_NULL_SIZE
- VID_HSA_TIME
- VID_HBP_TIME
- VID_HLINE_TIME
- VID_VSA_LINES
- VID_VBP_LINES
- VID_VFP_LINES
- VID_VACTIVE_LINES

For more information on registers related to Video Mode Pattern Generator, see “Register Descriptions” chapter.

2.10.3 Color Bar Pattern

The color bar pattern comprises eight bars of the colors white, yellow, cyan, green, magenta, red, blue, and black. To calculate the color width, divide the line pixel size (vertical pattern) or the number of lines (horizontal pattern) by eight.

In the vertical color bar mode, each single color bar has a width of the number of pixels in a line divided by eight. In case, the number of pixels in a line is not divisible by eight, the last color (black) contains the remaining.

In the horizontal color bar mode, each color line has a color width of the number of lines in a frame divided by eight. In case, the number of lines in a frame is not divisible by eight, the last color (black) contains the remaining lines.

Figure 2-40 Vertical Color Bar Mode

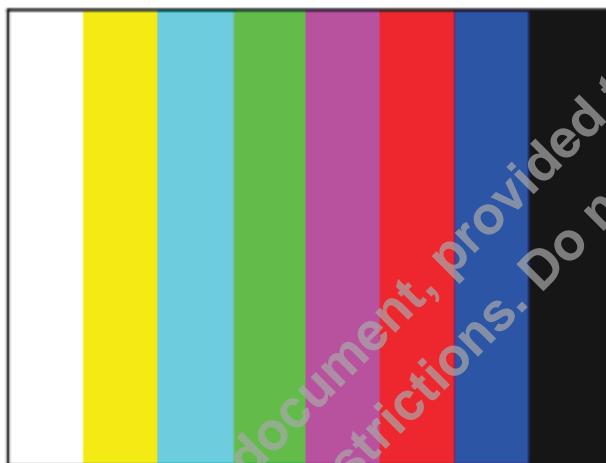


Figure 2-41 Horizontal Color Bar Mode



2.10.4 Color Coding

2.10.4.1 RGB Components

Table 2-5 shows the RGB components used.

Table 2-5 RGB Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
R	High	High	Low	Low	High	High	Low	Low
G	High	High	High	High	Low	Low	Low	Low
B	High	Low	High	Low	High	Low	High	Low

2.10.4.2 YCbCr Components

To avoid YCbCr negative components, YCbCr color coding is coded with following offsets:

- Y has a range of 219 and an offset of +16
- CB and CR have a range of ±112 and offset of +128

Table 2-6 shows YCbCr 8-Bit components.

Table 2-6 YCbCr 8-Bit Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
Y [7:0]	235	210	170	145	106	81	41	16
Cr [7:0]	128	16	166	54	202	90	240	128
Cb [7:0]	128	146	16	34	222	240	110	128

Table 2-7 shows YCbCr 10-Bit Components. Each component of the 20-Bit YCbCr 4:2:2 loosely packed is calculated by multiplying the respective component of the 16-Bit YCbCr 4:2:2 by 4.

Table 2-7 YCbCr 10-Bit Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
Y [9:0]	940	840	680	580	424	324	164	64
Cr [9:0]	512	64	664	216	808	360	960	512
Cb [9:0]	512	534	64	136	888	960	440	512

Table 2-8 shows YCbCr 12-Bit Components. Each component of the 24-Bit YCbCr 4:2:2 loosely packed is calculated by multiplying the respective component of the 16-Bit YCbCr 4:2:2 by 16.

Table 2-8 YCbCr 12-Bit Components

	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
Y [11:0]	3760	3360	2720	2320	1696	1296	656	256
Cr [11:0]	2048	256	2656	864	3232	1440	3840	2048
Cb [11:0]	2048	2336	256	544	3552	3840	1760	2048

2.10.5 BER Testing Pattern

The BER testing pattern simplifies conformance testing. This pattern tests the RX PHY capability to receive the data correctly. The following data patterns are required:

- X bytes of 0xAA (high-frequency pattern, inverted)
- X bytes of 0x33 (mid-frequency pattern)
- X bytes of 0xF0 (low-frequency pattern, inverted)
- X bytes of 0x7F (lone 0 pattern)
- X bytes of 0x55 (high-frequency pattern)
- X bytes of 0xCC (mid-frequency pattern, inverted)
- X bytes of 0x0F (low-frequency pattern)
- Y bytes of 0x80 (lone 1 pattern)

In most cases, Y is equal to X. However, depending on line length and the color coding used, Y may be a different value than X.

With RGB888 color coding and horizontal resolution in multiples of eight, the following pattern appears on the DSI display:

Figure 2-42 RGB888 BER Testing Pattern

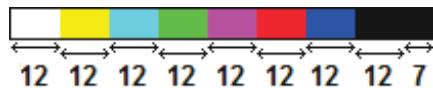


2.10.6 Video Mode Pattern Generator Resolution

The minimum supported resolutions for video pattern generator is resolution where VID_PKT_SIZE is greater than 32-bit.

2.10.6.1 Vertical Pattern

The division of horizontal resolution (pixels) for the eight test pattern colors, determines the width of each color. If the horizontal resolution is not divisible by eight, the last color (black) is extended to fill the resolution. In the example in [Figure 2-43](#), the horizontal resolution is 103.

Figure 2-43 Vertical Pattern (103*15)

2.10.6.2 Horizontal Pattern

The division of the number of vertical resolution (lines) for eight test pattern colors, determines the width of each color bar. If the vertical resolution is not divisible by eight, the last color (Black) is extended to fill the resolution.

Figure 2-44 Horizontal Pattern (103*15)

2.11 Display Stream Compression

The DWC_mipi_dsi_host controller supports the following compression modes:

- **DSC24:** The DWC_mipi_dsi_host controller handles incoming DPI data as compressed data, and therefore does not perform data compression.
- **DSC Encoder:** The DWC_mipi_dsi_host controller handles incoming DPI data as pixel data, and therefore compresses it according to the DSC encoder configuration.

In the DSC Encoder compression mode, the DWC_mipi_dsi_host controller supports up to two DSC encoders. Using two DSC encoders (dual DSC feature), provides the flexibility of selecting the desired architecture. Each architecture has its advantages and disadvantages in terms of area, number of required RAMs, and Clock Domain Crossing (CDC) structures.

To summarize, the available architectures are as follows:

- **Single DSC:** This is the most basic architecture with only one DSC instance available. By default, this is the architecture used when the DSC feature is active in coreConsultant. It compresses one slice at a time. Of all the available architectures, this requires the least number of RAMs. [Figure 2-45](#) illustrates this.
- **Dual DSC Single Port:** This architecture uses two DSC encoder instances which enables the possibility of processing two DSC slices in parallel. The incoming DPI pixel data is processed at $dipi\text{clk}/2$ rate by each DSC Instance. When compared to the Single DSC architecture, there is an increase the number of RAMs required and a significant increase in the area of the DWC_mipi_dsi_host controller. Besides the extra DSC related RAMs such as the line buffer and syntax elements RAMs, it also requires the slice demultiplexer RAM. Because this RAM is read at $dipi\text{clk}/2$ rate, both encoders can share it. On the other hand, the DPI/rate buffer RAM is written at $dipi\text{clk}$ rate, so, both encoders can share the existing DPI/rate buffer RAM (only its capacity must be increased). [Figure 2-46](#) illustrates this.
- **Dual DSC Single Port Dual Pixel:** This architecture also instantiates two DSC encoders. At each $dipi\text{clk}$ cycle, the DPI data bus contains the data of two contiguous pixels. The DSC instances are also clocked at $dipi\text{clk}$. This architecture has the advantage of not requiring the $dipi\text{clk}/2$ clock domain. The disadvantage is that it requires two slice demultiplexer RAMs and two DPI/rate buffer RAMs. [Figure 2-47](#) illustrates this.
- **Dual DSC Dual Port:** Similar to the dual DSC single port dual pixel architecture but there are two DPI data bus, each one carrying the pixel data of the corresponding horizontal slice instead of a single DPI data bus carrying two pixels per clock cycle. The advantage of this approach is that, this does not require the logic that separates the pixels for each encoder (slice demultiplexer). [Figure 2-48](#) illustrates this.

Figure 2-45 Single DSC Architecture (Simplified View)

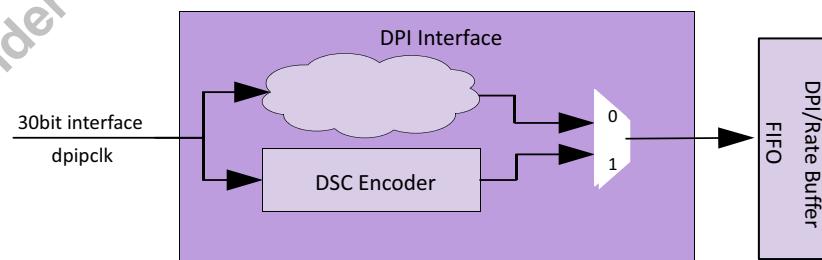


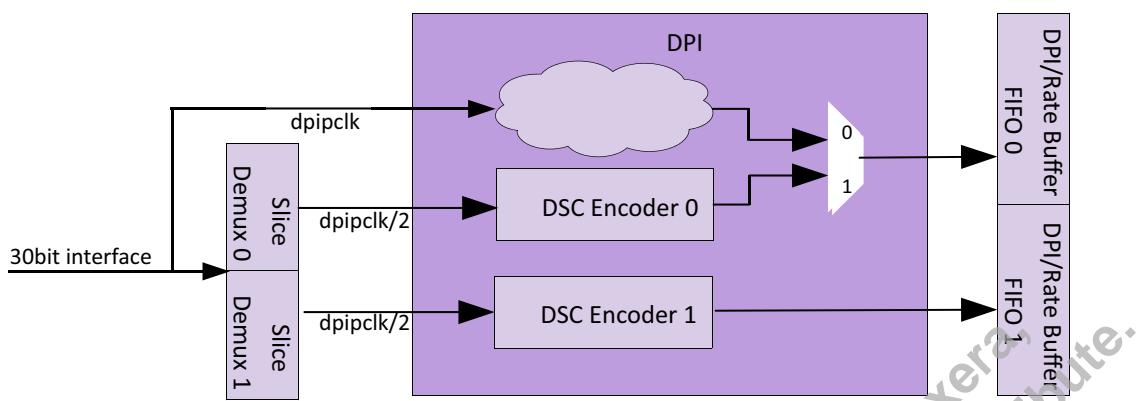
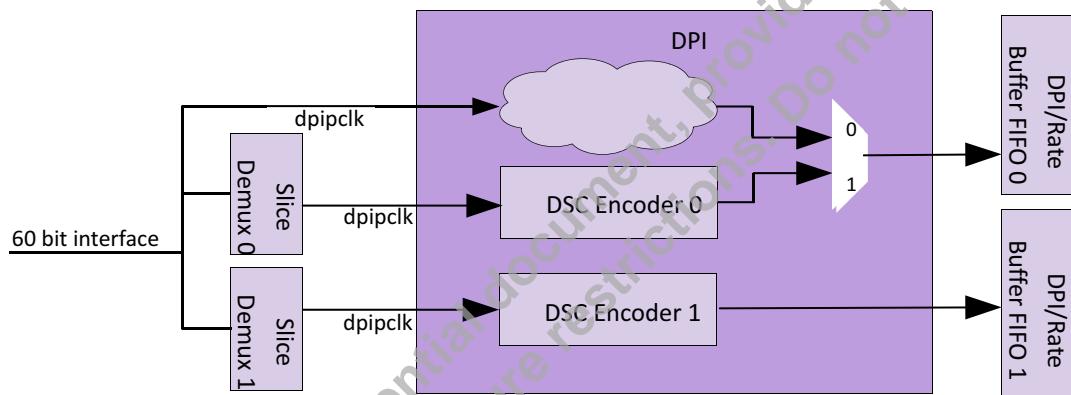
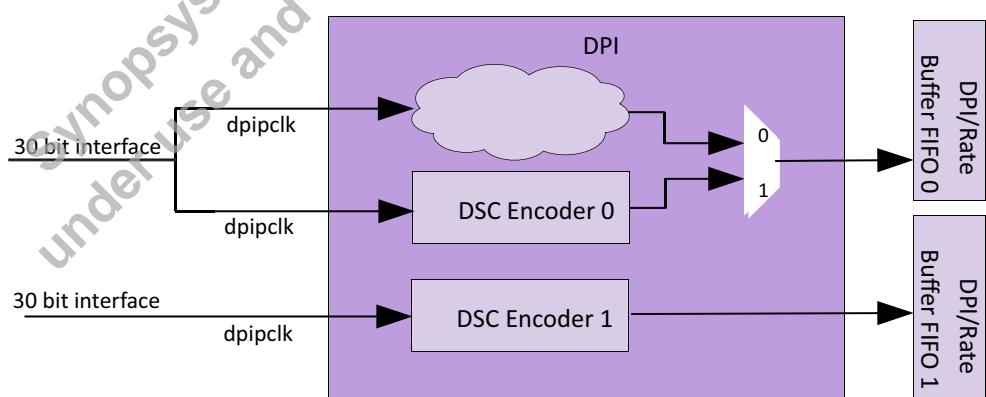
Figure 2-46 Dual DSC - Single Port Architecture (Simplified View)**Figure 2-47 Dual DSC - Single Port Dual Pixel Architecture (Simplified View)****Figure 2-48 Dual DSC Dual Port Architecture (Simplified View)**

Table 2-9 summarizes the single or dual DSC architecture information in terms of number of RAMs required, relative area, and clock domains.

Table 2-9 DSC Architectures (Continued)

	Default	Number of RAMs	Area	DPI Processing	Additional dpipclk Requirement	Horizontal Slices Max
Single DSC	Yes	5 (+1 shared) ¹	X	dpipclk	No	1
Dual DSC Single Port	No	11 (+1 shared) ¹	2X++	dpipclk/2	Yes - dpipclk/2	2
Dual DSC Single Port Dual pixel	No	13 (+1 shared) ¹	2X+++	dpipclk	No	2
Dual DSC Dual Port	No	11 (+1 shared) ¹	2X+	dpipclk	No	2

1. RAM that is shared by the rate buffer of the first instance of DSC encoder and DPI pixel FIFO of the DWC_mipi_dsi_host controller.

2.11.1 DSC Rate Buffer RAM Requirements

To avoid underflow errors, the DSC encoder must store some compressed lines in the Rate Buffer during the initial delay period, before data starts to be processed by the DSI protocol. Because of this reason, the DSC Rate Buffer size strongly depends on the number of delay lines.

The default value for the DSC delay parameter assumes that the entire slice must be stored in the rate buffer. This is the safest approach but has a significant impact in the memory requirements.

The default ram requirements are determined as follows:

- Rate buffer RAM width = 64 bits
- Default frame width (active video) = 1280 pixels
- Default frame height (active video) = 720 lines
- Default number of vertical slices = 1
- Default maximum bits per pixel (bpp) at the encoder output = 10 bpp
- Maximum slice height = frame height / number vertical slices = 720
- Number of delay lines = slice height (entire slice stored) = 720
- Rate buffer size (bits) = number delay lines * pixels per line * bpp = 9 216 000 bits
- Rate buffer size (depth) = ceil(rate buffer size (bits) / RAM Width) = 144 000 addresses
- Actual RAM size might require up to two extra addresses, relatively to the Rate buffer size (depth) value.

Using a dual DSC encoder configuration, requires a second rate buffer:

- 2nd rate buffer size (bits) = number delay lines * (pixels per frame line / 2) * bpp = 4 608 000 bits
- 1st rate buffer size (bits) = maintains the same size to support the single encoder scenario.

The previous discussion shows the importance of choosing reasonable values for the minimum number of vertical slices, maximum bpp and number of delay lines parameters.

The maximum bpp at the encoder output and the minimum number of vertical slices parameters, are straightforward to select based on the user requirements. The recommendation for the number of delay lines is:

Table 2-10 6bpp at The Encoder Output

Single Encoder or Dual Encoder in Single Slice Mode						Dual Encoder in Dual Slice Mode							
Frame width (range)	[320:420]	[420:520]	[520:680]	[680:980]	[980:11260]	1260 or greater	[320:460]	[460:560]	[560:760]	[760:960]	[960:1280]	[1280:1920]	1920 or greater
Recommended delay measured in lines	12	10	8	6	4	3	21	18	14	10	8	6	4

Table 2-11 8bpp at The Encoder Output

Single Encoder or Dual Encoder in Single Slice Mode				Dual Encoder in Dual Slice Mode						
Frame width (range)	[320:540]	[540:800]	[800:1800]	1800 and greater	[320:520]	[250:660]	[660:840]	[840:1760]	1760 and greater	
Recommended delay measured in lines	6	5	4	3	8	7	6	5	4	

The recommendation is two lines of delay for both configurations (single encoder and dual encoder).

These values are obtained by simulation using different images, with several slice width values tested for each image.



The number of delay lines also impacts the size of the buffer that stores DPI events (rising edge and falling edge of hsync, vsync, dataen, dpicolorm and dpishutdn signals).

Each time a signal toggles, an event is generated and is stored in the buffer (is a pulse constituted by a rising and a falling edge generates two events).

The buffer must be able to hold the events generated during the delay period.

2.11.2 Enabling Compression

It is possible to configure some display stream compression parameters using the compression mode command packet. This packet is automatically sent when an APB write is done in the DSC_PARAMETER register. However, it is also possible to send this packet manually using the APB Slave interface. For more information, see the “Sending Compressed Data” in the Programming chapter of the *DWC_mipi_dsi_host controller User Guide*.

To use DSC encoder, ensure that DSC_ENC_EN[0] & DSC_PARAMETER[0] are both asserted. **Table 2-12** represents the DSC encoder status according to these two parameters:

Table 2-12 Truth Table for DSC Encoder Enabling

DSC_ENC_EN[0]	DSC_PARAMETER[0]	DSI Host Output Data Type
0	X	Depending on the selected DPI_COLOR_CODING
1	0	Depending on the selected DPI_COLOR_CODING
1	1	Compressed pixel stream (0x0b)



You require a valid license to select DSC encoder feature.

2.11.3 DSC 24 - External DSC Encoder

To use this mode, DPI_COLOR_CODING register should be configured with DSC24 compressed data option. In this case, up to 3 bytes per clock cycle can be received. VID_PKT_SIZE and EDPI_CMD_SIZE register considers bytes and not pixels.

When using non-burst modes, each DSC chunk must start on a new DPICLK cycle. The DWC_mipi_dsi_host controller does not distinguish the differences between chunks and slices.

The two different scenarios are as follows:

- “Scenario 1: 1 Chunk Per Slice” on page [96](#)
- “Scenario 2: 1” on page [97](#)

2.11.3.1 Scenario 1: 1 Chunk Per Slice

[Figure 2-49](#) and [Figure 2-50](#) illustrate 1 chunk per slice.

Figure 2-49 3-Byte Chunks

DPI-2 Pixel Interface																														
	d29	d28	d27	d26	d25	d24	d23	d22	d21	d20	d19	d18	d17	d16	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Cycle1							Slice 1 Byte1						Slice 1 Byte2						Slice 1 Byte3											
Cycle2							Slice 2 Byte1						Slice 2 Byte2						Slice 2 Byte3											
Cycle3							Slice 3 Byte1						Slice 3 Byte2						Slice 3 Byte3											

Figure 2-50 4-Byte Chunks

DPI-2 Pixel Interface																														
	d29	d28	d27	d26	d25	d24	d23	d22	d21	d20	d19	d18	d17	d16	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Cycle1							Slice 1 Byte1						Slice 1 Byte2						Slice 1 Byte3											
Cycle2							Slice 2 Byte1						Slice 2 Byte2						Slice 2 Byte3											
Cycle3							Slice 3 Byte1						Slice 3 Byte2						Slice 3 Byte3											
Cycle4							Slice 2 Byte4																							
Cycle5							Slice 3 Byte1						Slice 3 Byte2						Slice 3 Byte3											
Cycle6							Slice 3 Byte4																							

2.11.3.2 Scenario 2: 1

Figure 2-51 6-Byte Chunks

		DPI-2 Pixel Interface																													
		d29	d28	d27	d26	d25	d24	d23	d22	d21	d20	d19	d18	d17	d16	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Cycle1								Slice 1 Byte1						Slice 1 Byte2						Slice 1 Byte3											
Cycle2								Slice 2 Byte1						Slice 2 Byte2						Slice 2 Byte3											
Cycle3								Slice 3 Byte1						Slice 3 Byte2						Slice 3 Byte3											
Cycle4								Slice 4 Byte1						Slice 4 Byte2						Slice 4 Byte3											

Figure 2-52 8-Byte Chunks

		DPI-2 Pixel Interface																													
		d29	d28	d27	d26	d25	d24	d23	d22	d21	d20	d19	d18	d17	d16	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
Cycle1								Slice 1 Byte1						Slice 1 Byte2						Slice 1 Byte3											
Cycle2								Slice 1 Byte4						Slice 2 Byte1						Slice 2 Byte2											
Cycle3								Slice 2 Byte3						Slice 2 Byte4																	
Cycle4								Slice 3 Byte1						Slice 3 Byte2						Slice 3 Byte3											
Cycle5								Slice 3 Byte4						Slice 4 Byte1						Slice 4 Byte2											
Cycle6								Slice 4 Byte3						Slice 4 Byte4																	

2.11.4 DSC Encoder - Internal Compression

On selecting VESA DSC encoder option in the coreConsultant, use the DSC_MODE register to enable this mode. In this case, VID_PKT_SIZE and EDPI_CMD_SIZE are programmed in units of pixel, according to the selected color coding.

The use cases that must be distinguished, are as follows:

- First use case: Uses only a single encoder instance, and therefore, processes only one slice at a time.
- Second use case: Uses two DSC encoders, and each one processing a different slice in parallel.

2.11.4.1 Single Encoder

Depending on the DSC encoder compression rate, some filler bits can be added to respect byte alignment required by the DSI specification. The filler bits are added, when needed, only at the end of each slice, since this allows DSC decoder to receive a continuous compressed data stream without discarding bits, except the last ones.

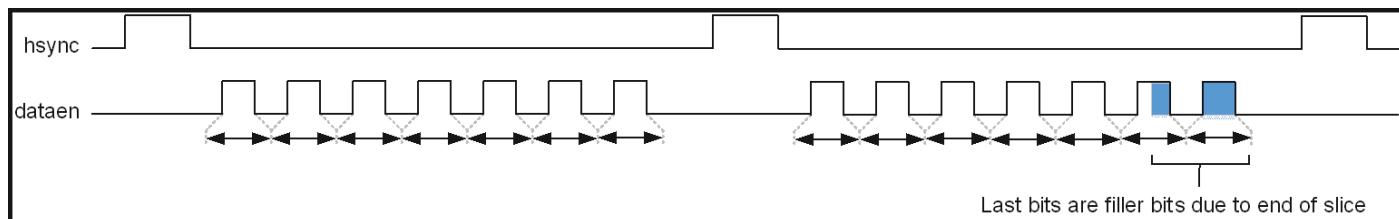
Figure 2-52 illustrates an example where the DWC_mipi_dsi_host controller adds filler bits to fill the required configuration:

- DSI Host controller configuration:
 - Video Mode – Non-Burst mode
 - VID_PKT_SIZE = 1 pixel
 - VID_NUM_CHUNKS = 7
 - DSC BPP = 12
 - DSC SLICE HEIGHT = 2 lines
- Number of bits per chunk output by the DWC_mipi_dsi_host controller:
 - (VID_PKT_SIZE*DSC_BPP rounded up to the next byte boundary) = 16
- Number of filler bits added by the DWC_mipi_dsi_host controller:

$$\square ((16 * \text{VID_NUM_CHUNKS}) - (\text{DSC_BPP} * \text{VID_NUM_CHUNKS})) = 28$$

Figure 2-53 represents this example in an entire slice, where blue represents the filler bits.

Figure 2-53 Example in The Line



To avoid the addition of filler bits, choose DSC BPP multiples of the byte.

2.11.4.2 Dual Encoder

When the Dual encoder feature is present and only one of the encoders is used, then the section “[DSC Encoder - Internal Compression](#)” on page 97 holds valid. For slice width half of the frame width, the two encoder instances produce a valid bit stream, that is multiplexed as a chunk at a time, from each encoder. Therefore, filler bytes, if required, are only added after processing the two horizontal slices. In order to ensure correct operation, the DWC_mipi_dsi_host controller must be configured to take into account the following items:

- Size of a DSC chunk (DSC CHUNK SIZE)
(SLICE WIDTH * DSC BPP rounded up to the next byte boundary)
- Number of bytes of a compressed frame line outputted by the two encoders (each encoder processes half of the frame line and contributes with DSC CHUNK SIZE bytes)
 $2 * \text{DSC CHUNK SIZE}$
- Number of bytes per DSI chunk outputted by DWC_mipi_dsi_host:
 $\text{DSI CHUNK BYTES} = (\text{VID_PKT_SIZE} * \text{DSC_BPP} \text{ rounded up to the next byte boundary})$
- Number of bytes per frame line outputted by DWC_mipi_dsi_host:
 $\text{FRAME LINE BYTES} = \text{DSI CHUNK BYTES} * \text{VID_NUM_CHUNKS}$
- $\text{FRAME LINE BYTES} \geq 2 * \text{DSC CHUNK SIZE}$

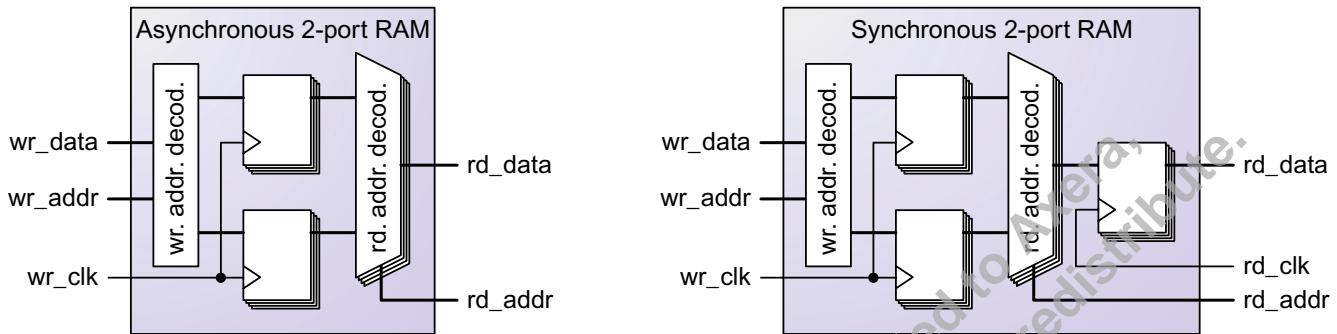


If $\text{FRAME LINE BYTES} = 2 * \text{DSC CHUNK SIZE}$, then no filler bytes are added.

2.12 Supported 2-Port RAM Types

The DWC_mipi_dsi_host controller uses 2-port RAMs to implement FIFOs for data storage and synchronization. Based on the way in which data is accessed for read operations, the 2-port RAMs are divided into two categories: synchronous and asynchronous. [Figure 2-54](#) illustrates this.

Figure 2-54 2-port RAM Types



In most of the cases, data can be read from an asynchronous 2-port RAM without a clock. Given a value for `rd_addr`, a word of data is selected for a read operation. It becomes available in `rd_data` as soon as it propagates across the selection logic. Because this data has to be captured by a read clock, it is necessary to know if the data becomes stable in time.

For a synchronous 2-port RAM, a read clock is needed to get the addressed read data. Data is retrieved synchronously with `rd_clk`, but at the expense of an extra clock cycle of delay.

Typically, asynchronous 2-port RAMs are slower than their synchronous counterparts. To capture its output read data safely in an external register, the read clock cannot be set as fast as it can be set for the synchronous read RAM. On the other hand, a synchronous 2-port RAM returns data at the interface one clock cycle after the read address is given.

The DWC_mipi_dsi_host controller works with both kinds of RAMs, yielding the same performance. The choice of RAM to use depends on the availability from the foundry and your preference. During the controller configuration in coreConsultant, use Select the type of 2-port RAM memory option to select the RAM type.

2.13 Timeout Timers

2.13.1 Peripheral Response Timeout

A peripheral may not immediately respond correctly to some received packets. For example, consider a peripheral receiving a read request, but due to its architecture, cannot access the RAM for a while. This may be because the panel is being refreshed, and therefore the peripheral takes some time to respond. In this case, set a timeout to ensure that the host waits long enough, so that the device processes the previous data before receiving the new data or responds correctly to new requests.

[Table 2-13](#) lists the events belonging to various categories having an associated timeout for peripheral response.

Table 2-13 List of Events of Different Categories of the PRESP_TO Counter

Category	Event
Items implying a BTA PRESP_TO	Bus Turn-Around
READ requests indicating a PRESP_TO (replicated for high-speed and low-power)	<ul style="list-style-type: none"> ■ (0x04) Generic read, no parameters short ■ (0x14) Generic read, 1 parameter short ■ (0x24) Generic read, 2 parameters short ■ (0x06) DCS read, no parameters short
WRITE requests indicating a PRESP_TO (replicated for high-speed and low-power)	<ul style="list-style-type: none"> ■ (0x03) Generic short write, no parameters short ■ (0x13) Generic short write, 1 parameter short ■ (0x23) Generic short write, 2 parameters short ■ (0x29) Generic long write long ■ (0x05) DCS short write, no parameters short ■ (0x15) DCS short write, 1 parameter short ■ (0x39) DCS long write/write_LUT Command packet long ■ (0x37) Set maximum return packet size

The DWC_mipi_dsi_host controller ensures that, on sending an event that triggers a timeout, the PHY switches to the Stop state and a counter starts running until it reaches the value of that timeout. The link remains in the low-power-IDLE state and unused until the timeout ends, even if there are other events ready to be transmitted.

[Figure 2-55](#), [Figure 2-56](#), and [Figure 2-57](#) illustrates the counting flow in the PRESP_TO counter for the three categories listed in [Table 2-13](#).

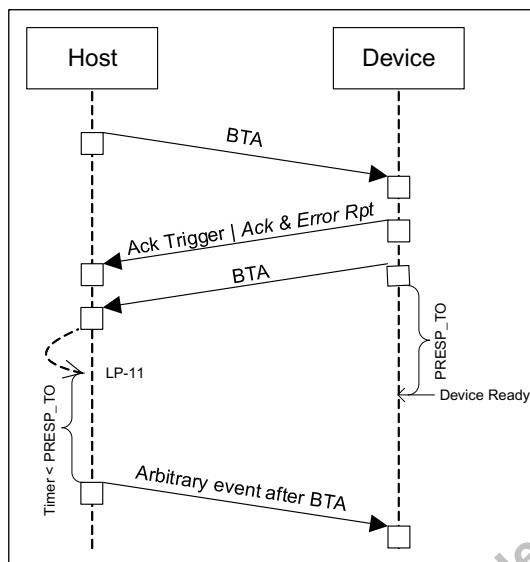
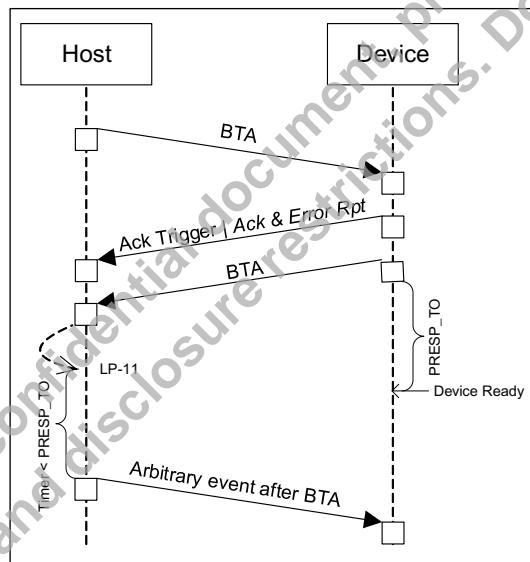
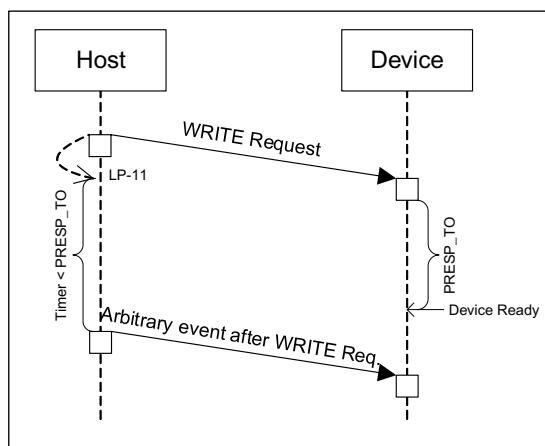
Figure 2-55 Timing of PRESP_TO after a Bus Turn-Around**Figure 2-56 Timing of PRESP_TO after a Read Request (high-speed or low-power)**

Figure 2-57 Timing of PRESP_TO after a Write Request (high-speed or low-power)



The BTA_TO_CNT, HS_RD_TO_CNT, LP_RD_TO_CNT, HS_WR_TO_CNT, and LP_WR_TO_CNT registers, configures the PRESP_TO counter.

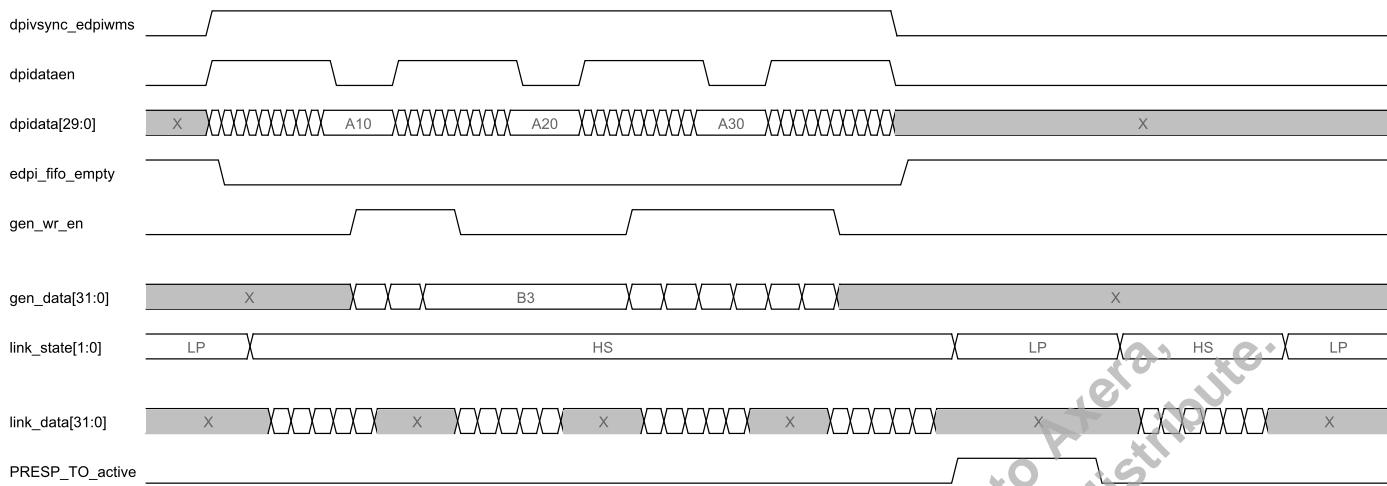
The values in these registers are measured in number of cycles of the lanebyteclk clock. Use these registers only in Command mode, because in Video mode, for proper refreshing of the display, a rigid timing schedule must be met and it must not be broken by these or any other timeouts. Setting a given timeout to 0, disables going into low-power-11 state, and timeout for events of that category.

The read and the write requests in high-speed mode are distinct from the read and the write requests in low-power mode. For example, on setting HS_RD_TO_CNT register to zero, and LP_RD_TO_CNT register to a non-zero value, then a generic read with no parameters does not activate the PRESP_TO counter in high-speed, but it activates the PRESP_TO counter in low-power.

The HS_RD_TO_CNT register includes a special bit, `presp_to_mode`, to change the normal behavior of PRESP_TO for the eDPI. When set to 1, this bit allows the PRESP_TO from HS_RD_TO_CNT to be used only once, when both of the following conditions are met:

- The `dpivsync_edpiwms` signal rises and falls.
- The packets originated from the eDPI are transmitted and its FIFO is empty again.

In this scenario, non-eDPI requests are not sent to the PHY, even if there is traffic from the Generic or the DBI ready to be sent, returning them to the Stop state. This activates the PRESP_TO counter and only when it is completed, the DWC_mipi_dsi_host controller, sends any other traffic that is ready. [Figure 2-58](#) illustrates this.

Figure 2-58 Effect of presp_to_mode at 1

This procedure has the following limitations:

- The edpiwms must remain de-asserted for at least 3 clock cycles before starting a new transaction.
- After de-asserting dpidataen, edpiwms must be de-asserted within the maximum latency calculated as follows:

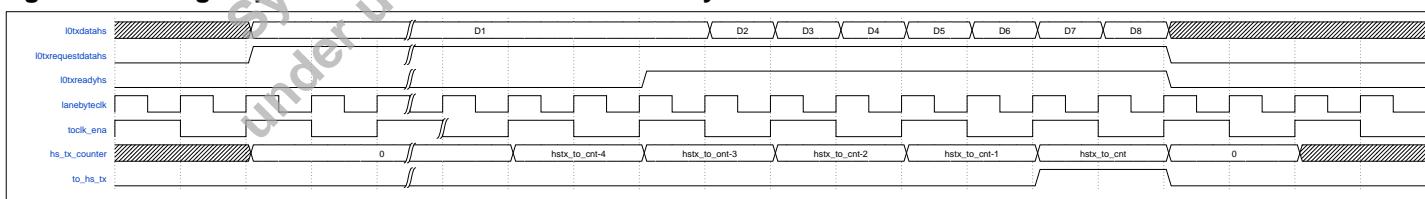
$$\text{Max_latency} = \text{lp2hs_time} + (\text{EDPI_CMD_SIZE} * \text{bytes_per_pixel} + 1 + 6) / \text{Number_of_lanes}$$

2.13.2 Errors Raised by Timeouts

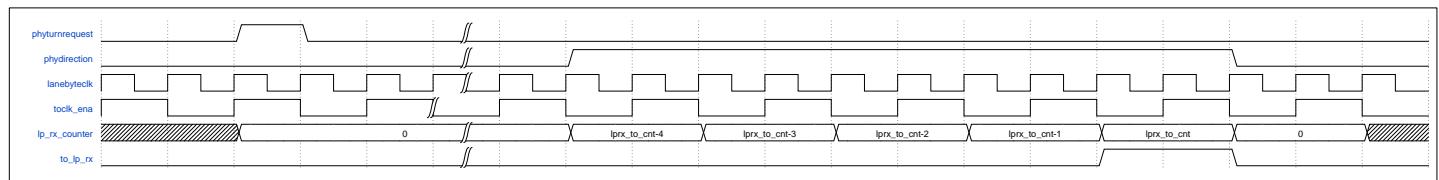
The DWC_mipi_dsi_host controller implements a set of timers and conditions to notify the errors. The DWC_mipi_dsi_host controller has a set of registers to control the timers used to determine if a timeout has occurred. It also contains a set of interruption status registers that are cleared upon a read operation.

Optionally, these registers also trigger an interrupt signal when an error occurs within the DSI connection.

When a high-speed transmission timeout occurs, the `to_hs_tx` field of the `INT_ST1` register is set, rising an interrupt pin. The timeout is configured in the `hstx_to_cnt` field of the `TO_CNT_CFG` register. A 16-bit counter measures the time during which the high-speed mode is active. If that counter reaches the value defined by the `hstx_to_cnt` field, the `to_hs_tx` field is asserted and an internal soft reset is generated to the DWC_mipi_dsi_host controller.

Figure 2-59 High-Speed Transmission Errors Raised by Timeouts

The `to_lp_rx` field of the `INT_ST1` register is set to flag the contention detection when a low-power reception timeout occurs, rising an interrupt pin. The timeout is configured in the `lprx_to_cnt` field of the `TO_CNT_CFG` register. A 16-bit counter measures the time during which the low-power reception is active. (i.e. Rising edge of phyturnrequest signal until the falling edge of the phydirection signal).

Figure 2-60 Low-Power Reception Errors Raised by Timeouts

If that counter reaches the value defined by the **lpx_to_cnt** field, the **to_lp_rx** field is asserted and an internal soft reset is generated to the DWC_mipi_dsi_host controller.

Time units for these 16-bit counters are configured in cycles defined in the **to_clk_division** field in the **CLKMGR_CFG** register. The value written to the **to_clk_division** field defines the time unit for the timeout limits using the Lane byte clock (**lanebyteclk**) as input. This mechanism increases the range to define these limits.

2.14 Error Control

The `INT_ST0` and `INT_ST1` registers report error conditions, and trigger an interrupt pin to inform the system about the occurrence of errors.

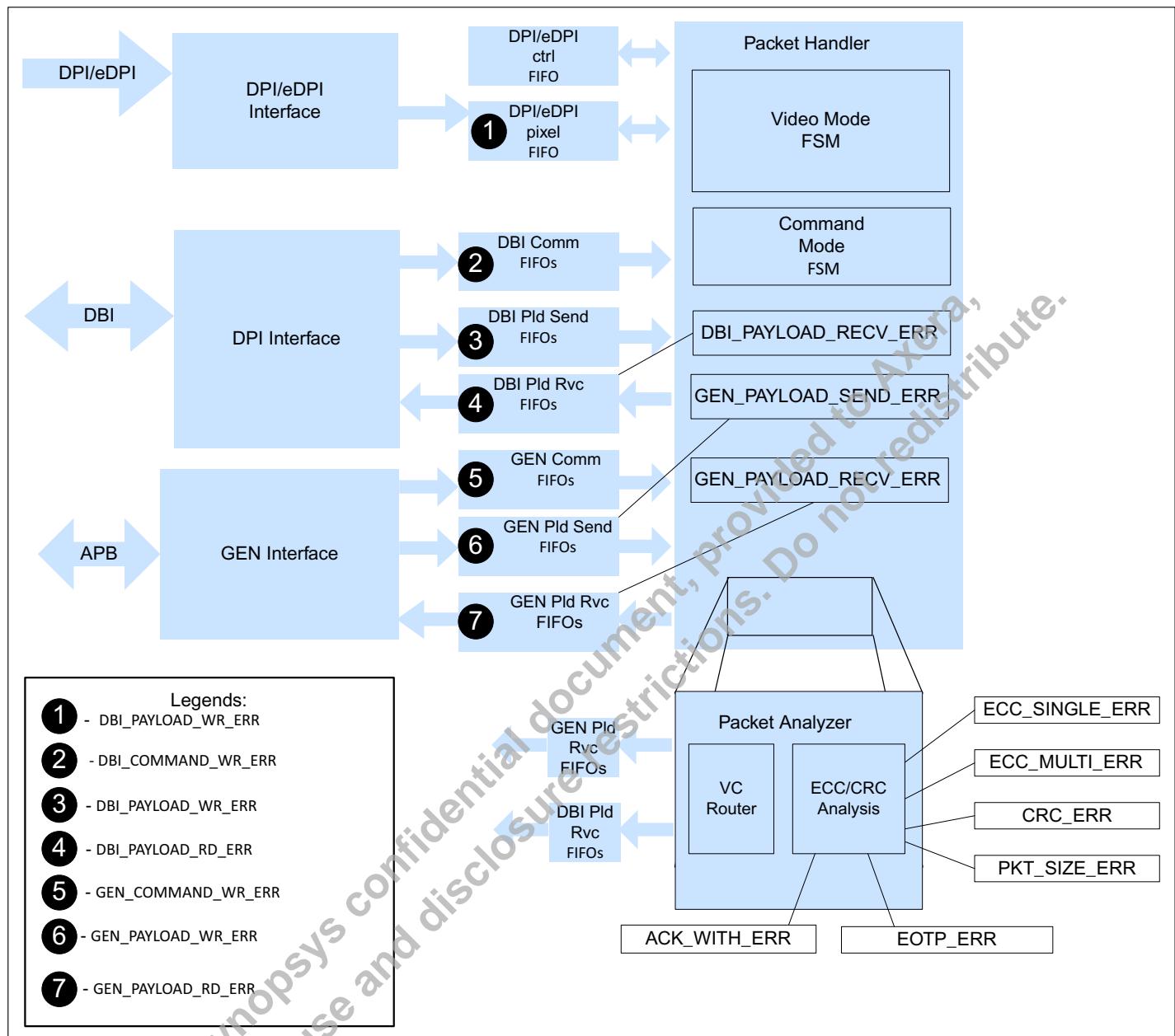
The `DWC_mipi_dsi_host` controller has one interrupt pin, that is set high when an error occurs in either the `INT_ST0` or the `INT_ST1` register.

Program the `INT_MSK0` and `INT_MSK1` mask registers, to mask the triggering of the interrupt pin. By default, all errors are masked. Setting any bit of these registers to 1, enables the interrupt for a specific error. The error bit is always set in the respective `INT_STn` register. The `INT_ST0` and `INT_ST1` registers are always cleared after a read operation. A read operation of all registers that caused the interrupt, clears the interrupt pin.

The interrupt force registers (`INT_FORCE0` and `INT_FORCE1`) are used for test purposes. These registers trigger the interrupt events individually without the need to activate the conditions that trigger the interrupt sources; this is because it is extremely complex to generate the stimuli for that purpose.

This feature also facilitates the development and testing of the software associated with the interrupt events. Setting any bit of these registers to 1 triggers the corresponding interrupt.

[Figure 2-61](#) on page 106 illustrates the location of some of the errors. For more information about the causes of the errors and the action to be taken, see “[Error Handling](#)” on page 429

Figure 2-61 Error Sources

Parameter Descriptions

This chapter details all the configuration parameters. You can use the coreConsultant GUI configuration reports to determine the complete configuration state of the controller. Some expressions might refer to TCL functions or procedures (sometimes identified as <functionof>) that coreConsultant uses to make calculations. The exact formula used by these TCL functions is not provided in this chapter. However, when you configure the controller in coreConsultant, all TCL functions and parameters are evaluated completely; and the resulting values are displayed where appropriate in the coreConsultant GUI reports.

The parameter descriptions in this chapter include the **Enabled:** attribute which indicates the values required to be set on other parameters before you can change the value of this parameter.

These tables define all of the configuration options for this component.

- “Core Configuration Parameters” on page 108
- “FIFOs Configuration Parameters” on page 110
- “VESA DSC Configuration Parameters” on page 115

3.1 Core Configuration Parameters

Table 3-1 Core Configuration Parameters

Label	Description
Select the Physical Interface	<p>Physical Interface. When Combo PHY is selected, the D-PHY has 4 lanes and C-PHY 3 lanes.</p> <p>License Dependencies: To generate a DWC_mipi_dsi_host with support for Combo C/D-PHY, you must have an associated DWC_mipi_dsi_host_combo license.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ D-PHY (1) ■ Combo_C-PHY_D-PHY (2) <p>Default Value: D-PHY</p> <p>Enabled: Always</p> <p>Parameter Name: DSI_HOST_PHY</p>
Select the number of lanes supported by the D-PHY	<p>Specifies the maximum number of data lanes in the design.</p> <p>Values: 1, 2, 3, 4</p> <p>Default Value: (DSI_HOST_PHY == 1) ? 2: 4</p> <p>Enabled: DSI_HOST_PHY == 1</p> <p>Parameter Name: DSI_HOST_DPHY_NUMBER_OF_LANES</p>
Use SNPS PHY	<p>Enables the Synopsys PHY as an internal submodule in the design. You need to have access to the Synopsys PHY package and set up the following variables:</p> <ul style="list-style-type: none"> ■ setenv DSI_HOST_PHY_PATH <install directory for PHY> ■ setenv DSI_HOST_PLL_PATH <install directory for PLL> ■ setenv DSI_HOST_PHY_LIBNAME <phyname.lib> ■ setenv DSI_HOST_PLL_LIBNAME <pllname.lib> <p>Values: 0, 1</p> <p>Default Value: 0</p> <p>Enabled: DSI_HOST_PHY==1</p> <p>Parameter Name: DSI_HOST_SNPS_PHY</p>

Label	Description
Select the system interface	<p>Configures the system data interface. Different memory models are required to support these interfaces.</p> <ul style="list-style-type: none"> ■ DBI: MIPI Display Bus Interface ■ DPI: MIPI Display Pixel Interface ■ DBI&DPI: Both MIPI Bus and Pixel interfaces ■ EDPI: Enhanced MIPI Display Pixel Interface <p>Values:</p> <ul style="list-style-type: none"> ■ DBI (1) ■ DPI (2) ■ DBI&DPI (3) ■ EDPI (4) <p>Default Value: EDPI</p> <p>Enabled: Always</p> <p>Parameter Name: DSI_HOST_DATAINTERFACE</p>
Select the DBI interface type	<p>Configures the type of DBI system interface. This option is available when you select DBI or DBI&DPI as the system interface (DSI_HOST_DATAINTERFACE = DBI or DBI&DPI).</p> <ul style="list-style-type: none"> ■ TYPE_A_FIXED_E: Type A interface in fixed E mode ■ TYPE_A_CLOCKED_E: Type A interface in clocked E mode ■ TYPE_B: Type B interface <p>This parameter is enabled only when DBI is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ TYPE_A_FIXED_E (0) ■ TYPE_A_CLOCKED_E (1) ■ TYPE_B (2) <p>Default Value: (DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)</p> <p>Enabled: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Parameter Name: DSI_HOST_DBINTERFACE_TYPE</p>
Select the number of synchronization stages	<p>Specifies the number of synchronization stages for clock domain crossing. All the stages capture the data on the rising edge of the clock.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 2 (2) ■ 3 (3) ■ 4 (4) <p>Default Value: 2</p> <p>Enabled: Always</p> <p>Parameter Name: DSI_HOST_DFLT_F_SYNC_TYPE</p>

3.2 FIFOs Configuration Parameters

Table 3-2 FIFOs Configuration Parameters

Label	Description
Select the type of 2-port RAM memory	<p>Configures the memory type to which the FIFO controllers connect.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ Asynchronous (0) ■ Synchronous (1) <p>Default Value: Synchronous</p> <p>Enabled: Always</p> <p>Parameter Name: DSI_HOST_SNPS_SYNC_RD_FIFOS</p>
DPI Payload Size	<p>Specifies DPI Payload size, in bytes. Define the DPI Payload Size such that it is sufficient to consider at least one entire line of video with the maximum resolution that you want to support.</p> <p>For example, consider supporting Full HD 1080p, where one video line contains 1920 pixels. If you use 24 bpp, each line contains $(1920 \times 24)/8 = 5760$ bytes.</p> <p>For loosely packed data types, while calculating the storage requirements, it is necessary to consider the bits at 0 that are transmitted as part of that data types. Therefore, for loosely packed 18-bit and 20-bit YCbCr 4:2:2 data types, the memory occupancy is 24 bits per pixel.</p> <p>Note: The eDPI interface also shares the DPI FIFO and so, the selected DPI FIFO depth also affects the eDPI operation. However, the eDPI interface has the capability of defining the packet transmission sizes. Therefore, the eDPI interface can always be configured to operate properly with that size irrespective of the size configured for the DPI.</p> <p>Values: (DSI_HOST_DPIINTERFACE == 1? 32:0), ..., 16384</p> <p>Default Value: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1) ? 1952 : 0</p> <p>Enabled: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)))</p> <p>Parameter Name: DSI_HOST_DPI_PLD_SIZE</p>
DPI Payload Required FIFO RAM Depth	<p>Specifies the DPI Payload required FIFO 2-port RAM Depth. Based on the address depth required, the actual RAM size might require up to two extra positions:</p> <p>If the address depth is odd, then one extra RAM position is required,</p> <p>If the address depth is even but not an integer power of two, then two extra RAM positions are required,</p> <p>Otherwise no extra RAM positions are required.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: DSI_HOST_DSC_ENC ? DSI_HOST_DPI_RB_RAM_ADDRDEPTH : DSI_HOST_DPI_FIFO_ADDRDEPTH</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DPI_PLD_FIFO_RAMDEPTH</p>

Label	Description
DBI Command Size	<p>Specifies the DBI Command Size, in bytes. It requires a FIFO, implemented using flip-flops.</p> <p>Define the DBI Command Size with the maximum number of packets that are expected to be in the queue at any given time.</p> <p>For example, if the system uses a DMA based access that can write a burst of commands to the DBI interface and resume normal procedure, the DBI Command Size should be configured with the minimum size of that burst.</p> <p>Consider that the initialization sequence of a given display consists of 10 sequential commands. In this case, you should define a minimum of $10 * 32 / 8$ (40) for the DBI Command FIFO depth.</p> <p>Values: (DSI_HOST_DBINTERFACE == 1? 32:0), ..., 4096</p> <p>Default Value: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) == 1) ? 40 : 0</p> <p>Enabled: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Parameter Name: DSI_HOST_DBICMD_SIZE</p>
DBI Payload Size	<p>Specifies the DBI Payload Size, in bytes. It requires a 2-Port RAM model.</p> <p>Define the DBI Write Payload Size such that it is sufficient to store the entire payload content of the number of packets defined in the DBI Command Size.</p> <p>For example, if the DBI command burst consists of 10 commands, and each command has an estimated mean payload of 40 bytes, then the total payload size should be $10 \times 40 = 400$ bytes. Therefore, you should define a minimum of 400 bytes for DBI Payload Size.</p> <p>Values: (DSI_HOST_DBINTERFACE == 1? 196 : 0), ..., 4096</p> <p>Default Value: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) == 1) ? 200 : 0</p> <p>Enabled: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Parameter Name: DSI_HOST_DBIPLD_SIZE</p>
DBI Payload Required FIFO RAM Depth	<p>Depth of the DBI Payload 2-port RAM. If DSI_HOST_DBIPLDADDRDEPTH is odd, then the value of DSI_HOST_DBIPLD_RAM_ADDRDEPTH is equal to (DSI_HOST_DBIPLDADDRDEPTH + 1). If DSI_HOST_DBIPLDADDRDEPTH is an even value but not an integer power of two, then the value of DSI_HOST_DBIPLD_RAM_ADDRDEPTH should be configured to (DSI_HOST_DBIPLDADDRDEPTH + 2), otherwise it should be configured to DSI_HOST_DBIPLDADDRDEPTH.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: (DSI_HOST_DBINTERFACE == 1) ? [<functionof> DSI_HOST_DBIPLDADDRDEPTH] : 0</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DBIPLD_RAM_ADDRDEPTH</p>

Label	Description
DBI Read Size	<p>Specifies the DBI Read Size, in bytes. It requires a 2-Port RAM model. Define the DBI Read Payload Size such that it is sufficient to store the largest read data packet that can be received from the display. For example, consider that the largest read data packet that can be received from the display contains 40 bytes of data. Therefore, you should define a minimum of 40 bytes for DBI Read Size.</p> <p>Note: There are commands available to configure the maximum size of a read back packet payload size for a display device. Therefore, it is possible to configure the display device to transmit the read responses with the sizes that fit the configured memory.</p> <p>Values: (DSI_HOST_DBINTERFACE == 1? 8:0), ..., 1024 Default Value: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) == 1) ? 20 : 0 Enabled: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) Parameter Name: DSI_HOST_DBIREADPLD_SIZE</p>
DBI Read Required FIFO RAM Depth	<p>Depth of the DBI Read 2-port RAM. If DSI_HOST_DBIREADPLDADDRDEPTH is odd, then the value of DSI_HOST_DBIREADPLD_RAM_ADDRDEPTH is equal to (DSI_HOST_DBIREADPLDADDRDEPTH + 1). If DSI_HOST_DBIREADPLDADDRDEPTH is an even value but not an integer power of two, then the value of DSI_HOST_DBIREADPLD_RAM_ADDRDEPTH should be configured to (DSI_HOST_DBIREADPLDADDRDEPTH + 2), otherwise it should be configured to DSI_HOST_DBIREADPLDADDRDEPTH.</p> <p>Values: -2147483648, ..., 2147483647 Default Value: (DSI_HOST_DBINTERFACE == 1) ? [<functionof> DSI_HOST_DBIREADPLDADDRDEPTH] : 0 Enabled: 0 Parameter Name: DSI_HOST_DBIREADPLD_RAM_ADDRDEPTH</p>
Generic Command Size	<p>Generic Command FIFO Address Depth.</p> <p>Specifies the Generic Command Size, in bytes. It requires a FIFO, implemented using flip-flops.</p> <p>Define the Generic Command Size with the maximum number of packets that are expected to be in the queue at any given time. For example, if the system uses a DMA based access that can write a burst of commands to the Generic interface and resume normal procedure, the Generic Command Size should be configured with the minimum size of that burst.</p> <p>Consider that the initialization sequence of a given display consists of 10 sequential commands. In this case, you should define a minimum of 10*32/8 (40) for the Generic Command Size.</p> <p>Values: 32, ..., 4096 Default Value: 40 Enabled: Always Parameter Name: DSI_HOST_GENERICCMD_SIZE</p>

Label	Description
Generic Payload Size	<p>Specifies the Generic Payload Size, in bytes. It requires a 2-Port RAM model. Define the Generic Write Payload Size such that it is sufficient to store the entire payload content of the number of packets defined in the Generic Command Size. For example, if the Generic command burst consists of 10 commands, and each command has an estimated mean payload of 40 bytes, the total payload size should be $10 \times 40 = 400$ bytes. Therefore, you should define a minimum of 400 bytes for the Generic Write Payload Size.</p> <p>Values: 32, ..., 16384 Default Value: 800 Enabled: Always Parameter Name: DSI_HOST_GENERICPLD_SIZE</p>
Generic Payload Required FIFO RAM Depth	<p>Depth of the Generic Payload 2-port RAM.</p> <p>If DSI_HOST_GENERICPLDADDRDEPTH is odd, then the value of DSI_HOST_GENERICPLD_RAM_ADDRDEPTH is equal to (DSI_HOST_GENERICPLDADDRDEPTH + 1).</p> <p>If DSI_HOST_GENERICPLDADDRDEPTH is an even value but not an integer power of two, then the value of DSI_HOST_GENERICPLD_RAM_ADDRDEPTH should be configured to (DSI_HOST_GENERICPLDADDRDEPTH + 2), otherwise it should be configured to DSI_HOST_GENERICPLDADDRDEPTH.</p> <p>Values: -2147483648, ..., 2147483647 Default Value: [<functionof> DSI_HOST_GENERICPLDADDRDEPTH] Enabled: 0 Parameter Name: DSI_HOST_GENERICPLD_RAM_ADDRDEPTH</p>
Generic Read Size	<p>Specifies the Generic Read Size, in bytes. It requires a 2-Port RAM model. Define the Generic Read Payload Size such that it is sufficient to store the largest read data packet that can be received from the display.</p> <p>For example, consider that the largest read data packet that can be received from the display contains 40 bytes of data. Therefore, you should define a minimum of 40 bytes for the Generic Read Payload Size.</p> <p>Note: There are commands available to configure the maximum size of a read back packet payload size for a display device. Therefore, it is possible to configure the display device to transmit the read responses with the sizes that fit the configured memory.</p> <p>Values: 32, ..., 4096 Default Value: 64 Enabled: Always Parameter Name: DSI_HOST_GENREADPLD_SIZE</p>

Label	Description
Generic Read Required FIFO RAM Depth	<p>Depth of the Generic Read 2-port RAM. If DSI_HOST_GENREADPLDADDRDEPTH is odd, then the value of DSI_HOST_GENREADPLD_RAM_ADDRDEPTH is equal to (DSI_HOST_GENREADPLDADDRDEPTH + 1). If DSI_HOST_GENREADPLDADDRDEPTH is an even value but not an integer power of two, then the value of DSI_HOST_GENREADPLD_RAM_ADDRDEPTH should be configured to (DSI_HOST_GENREADPLDADDRDEPTH + 2), otherwise it should be configured to DSI_HOST_GENREADPLDADDRDEPTH.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: [<functionof> DSI_HOST_GENREADPLDADDRDEPTH]</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_GENREADPLD_RAM_ADDRDEPTH</p>

3.3 VESA DSC Configuration Parameters

Table 3-3 VESA DSC Configuration Parameters

Label	Description
Internal DSC encoder (require extra RAMs)	<p>When selected, Vesa DSC 1v1 Encoder is included in DWC_mipi_dsi_host. This require extra RAMs.</p> <p>License Dependencies: To generate a DWC_mipi_dsi_host with support for DSC encoder, you must have an associated DWC_mipi_dsi_host_dsc license.</p> <p>Values: 0, 1 Default Value: 0 Enabled: (DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 4) Parameter Name: DSI_HOST_DSC_ENC</p>
DSC Encoder Selection	<p>DSC Encoder Selection.</p> <ul style="list-style-type: none"> ■ Single DSC: One DSC encoder with a single Pixel Port ■ Dual DSC Single Port: This architecture uses two DSC encoder with a single Pixel Port ■ Dual DSC Single Port with Dual pixel: This architecture uses two DSC encoder with a Dual Pixel Port ■ Dual DSC Dual Port: This architecture uses two DSC encoder with two single Pixel Port <p>Values:</p> <ul style="list-style-type: none"> ■ Single_DSC (1) ■ Dual_DSC_Single_Port (2) ■ Dual_DSC_Single_Port_Dual_Pixel (3) ■ Dual_DSC_Dual_Port (4) <p>Default Value: Single_DSC Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSI_HOST_DSC_ENC_TYPE</p>
Maximum number of pixels per line	<p>Maximum number of pixels per line, supported by DSC encoder</p> <p>Values: 640, ..., 8192 Default Value: 1280 Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSC_MAX_PIXS_PER_LINE</p>
Maximum number of lines	<p>Maximum number of lines, supported by DSC encoder</p> <p>Values: 480, ..., 4320 Default Value: 720 Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSC_MAX_NUM_LINES</p>

Label	Description
Maximum value of bits per pixel at the encoder output	Maximum value of bits per pixel (bpp) at the encoder output. This value defines the minimum compression ratio. Values: <ul style="list-style-type: none"> ■ 6bpp (6) ■ 8bpp (8) ■ 10bpp (10) ■ 12bpp (12) ■ 15bpp (15) Default Value: 15bpp Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSI_HOST_DSC_MAX_BPP
Minimum number of vertical slices	Minimum number of vertical slices (Max slice_height = Maximum number of lines per frame / Minimum number of vertical slices) Values: <ul style="list-style-type: none"> ■ 1 (1) ■ 2 (2) ■ 3 (3) ■ 4 (4) Default Value: 1 Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSI_HOST_DSC_MIN_VSLICES
Initial DSC delay measured in lines	Number of lines of delay before compressed data starts being retrieved from the DSC Encoder. Values: -2147483648, ..., 2147483647 Default Value: [<functionof> {int(cell(DSC_MAX_NUM_LINES/DSI_HOST_DSC_MIN_VSLICES))}] Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSI_HOST_DSC_DELAY_LINES
Override recommended configuration	Override DSC RAMs configuration. If disabled, then the recommended values are used. Values: 0, 1 Default Value: 0 Enabled: DSI_HOST_DSC_ENC == 1 Parameter Name: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS
Maximum events to keep track during DSC delay	Maximum number of events to keep track of, for DPI delay. This value should be at least four times the DSC delay in lines (plus two lines of safety) = 4*(DSI_HOST_DSC_DELAY_LINES+2). Values: -2147483648, ..., 2147483647 Default Value: [<functionof> {(DSI_HOST_DSC_DELAY_LINES+2)*4}] Enabled: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS==1 Parameter Name: DSI_HOST_DPI_DELAY_MAX_EVENTS

Label	Description
DSC rate buffer FIFO Depth (uses 2-port RAM)	<p>Address depth of the Rate Buffer RAM for the first DSC Encoder instance. If the single DSC or the Dual Encoder Single Port architecture are used, then the DPI FIFO will also be used as a Rate Buffer by each DSC.</p> <p>In the remaining architectures, besides the DPI FIFO (Rate Buffer for the first instance of the DSC), an extra RAM serving as a Rate Buffer for the second instance, will be required. Configure as follows:</p> <ul style="list-style-type: none"> ■ If you intend to send compressed data on a per line basis, with a latency of 4 lines relatively to DPI interface, a safe value is the size of 6 compressed lines. ■ General rule is to use the size that allows the buffer to accommodate your intended latency, measured in compressed lines, plus 1 compressed line, for margin. <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: [<functionof> DSC_MAX_PIXS_PER_LINE 64 DSI_HOST_DSC_DELAY_LINES DSI_HOST_DSC_MAX_BPP]</p> <p>Enabled: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS==1</p> <p>Parameter Name: DSI_HOST_RB_ADDRDEPTH_FIRST_DSC</p>
DSC rate buffer FIFO Depth (Second DSC instance) (uses 2-port RAM)	<p>Address depth of the Rate Buffer RAM for the second DSC Encoder instance. Since the second DSC instance is only active when two slices per line are used, the depth requirements is half of the requirements for the first DSC instance.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: [<functionof> DSI_HOST_DSC_ENC_TYPE DSC_MAX_PIXS_PER_LINE 64 DSI_HOST_DSC_DELAY_LINES DSI_HOST_DSC_MAX_BPP]</p> <p>Enabled: (DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS==1) && (DSI_HOST_DSC_ENC_TYPE != 1)</p> <p>Parameter Name: DSI_HOST_RB_ADDRDEPTH_SEC_DSC</p>

Label	Description
DPI FIFO (DSC Rate Buffer) RAM depth	<p>Required DSI FIFO / DSC Rate Buffer ram depth based on DSC requirements for the Rate Buffer and DSI requirements for the DPI FIFO.</p> <p>Regarding the DSC requirements:</p> <p>If the DSC feature is active and the single encoder or the dual encoder single port architectures are used, then the rate buffer requirements for both encoders are considered because only one RAM serving as Rate Buffer / DSI FIFO is used. For the remaining dual encoder architectures only the requirements for the first DSC instance are considered because a second RAM is used as a Rate Buffer for the second DSC instance.</p> <p>Based on the address depth required, the actual RAM size might require up to two extra positions:</p> <p>If the address depth is odd, then one extra RAM position is required,</p> <p>If the address depth is even but not an integer power of two, then two extra RAM positions are required,</p> <p>Otherwise no extra RAM positions are required.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: (DSI_HOST_DPIINTERFACE == 0) ? 0 : [<functionof> DSI_HOST_DSC_ENC DSI_HOST_DSC_ENC_TYPE DSI_HOST_DPI_PLD_FIFO_ADDRDEPTH DSI_HOST_RB_ADDRDEPTH_FIRST_DSC DSI_HOST_RB_ADDRDEPTH_SEC_DSC]</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DPI_RB_RAM_ADDRDEPTH</p>
Second DSC Rate Buffer RAM depth	<p>Required Rate Buffer RAM depth for the second DSC instance.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: [<functionof> DSI_HOST_RB_ADDRDEPTH_SEC_DSC]</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_RB_SEC_DSC_RAM_ADDRDEPTH</p>
Line Buffer Ram Depth	<p>Line Buffer Ram Depth</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: DSC_MAX_PIXS_PER_LINE</p> <p>Enabled: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS == 1</p> <p>Parameter Name: DSC_LB_RAM_DEPTH</p>
Number of Line Buffer RAMs in the design	<p>Number of Line Buffer RAMs in the design.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: (DSI_HOST_DSC_ENC_TYPE == 1) ? 1 : 2</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DSC_LB_RAM_NINST</p>
Number of Rate Buffer RAMs in the design	<p>Number of Rate Buffer RAMs in the design</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: ((DSI_HOST_DSC_ENC_TYPE == 1) (DSI_HOST_DSC_ENC_TYPE == 2)) ? 1 : 2</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DSC_RB_RAM_NINST</p>

Label	Description
Balance RAM Depth	<p>Balance RAM Depth $DSC_BALANCE_RAM_DEPTH=\text{ceil}(DSC_BALANCE_FIFO_SIZE * 8 / DSC_BALANCE_RAM_DWIDTH)$ $DSC_BALANCE_FIFO_SIZE$ is determined by $((((DSC_MAX_MUX_WORD_SIZE + DSC_MAX_SE_SIZE - 1) * DSC_MAX_SE_SIZE + 7) / 8) + 2) + (DSC_MAX_SE_SIZE * 32) / 8$</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: =[<functionof> 10 32]</p> <p>Enabled: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS == 1</p> <p>Parameter Name: DSC_BALANCE_RAM_DEPTH</p>
Number of Balance RAMs in the design	<p>Number of Balance RAMs in the design.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: (DSI_HOST_DSC_ENC_TYPE == 1) ? 3 : 6</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DSC_BAL_RAM_NINST</p>
Syntax Element Size RAM depth	<p>Syntax Elements RAM Depth $DSC_SE_SIZE_RAM_DEPTH=\text{ceil}(DSC_SE_SIZE_FIFO_SIZE * 8)$ $DSC_SE_SIZE_FIFO_SIZE$ is given by $((6 * (DSC_MAX_MUX_WORD_SIZE + DSC_MAX_SE_SIZE - 1) + 54 * 8 + 31) / 32 * 4)$</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: =[<functionof> 10]</p> <p>Enabled: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS == 1</p> <p>Parameter Name: DSC_SE_SIZE_RAM_DEPTH</p>
Number of Syntax Element Size RAMs in the design	<p>Number of Syntax Element Size RAMs in the design.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: (DSI_HOST_DSC_ENC_TYPE == 1) ? 1 : 2</p> <p>Enabled: 0</p> <p>Parameter Name: DSI_HOST_DSC_SESIZE_RAM_NINST</p>
Slice demultiplexer depth per DSC instance	<p>Required Slice Demultiplexer RAM depth per DSC instance. For Dual DSC Single Port Dual pixel, this value should be equal to $\text{ceil}(DSC_MAX_PIXS_PER_LINE / 8) + 1$. For Dual DSC Single Port Single Pixel this value should be equal to $\text{ceil}(DSC_MAX_PIXS_PER_LINE / 4) + 1$.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: [<functionof> DSC_MAX_PIXS_PER_LINE DSI_HOST_DSC_ENC_TYPE]</p> <p>Enabled: DSI_HOST_OVERRIDE_DSC_RAM_CONFIGS == 1</p> <p>Parameter Name: DSC_SLICE_DEMUX_DEPTH_PER_INSTANCE</p>

Label	Description
Slice demultiplexer RAM depth	<p>Required Slice Demultiplexer RAM depth when the Dual DSC Encoder feature is active. If each DSC instance shares the same Demultiplexer RAM as in the "Dual DSC Single Port architecture", then the required RAM depth is given by $2 \times \text{DSC_SLICE_DEMUX_DEPTH_PER_INSTANCE}$. In the remaining Dual DSC architectures, two RAMs are required, each one with $\text{DSC_SLICE_DEMUX_DEPTH_PER_INSTANCE}$ positions. Notice: Up to two extra RAM addresses might be required per instance, relatively to the minimum requirements.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: [<functionof> $\text{DSC_SLICE_DEMUX_DEPTH_PER_INSTANCE}$ $\text{DSI_HOST_DSC_ENC_TYPE}$]</p> <p>Enabled: 0</p> <p>Parameter Name: <code>DSC_SLICE_DEMUX_RAM_DEPTH</code></p>
Number of Slice Demultiplexer RAMs in the design	<p>Number of Slice Demultiplexer RAMs in the design.</p> <p>Values: -2147483648, ..., 2147483647</p> <p>Default Value: $(\text{DSI_HOST_DSC_ENC_TYPE} == 1) \parallel (\text{DSI_HOST_DSC_ENC_TYPE} == 4) ? 0 : (\text{DSI_HOST_DSC_ENC_TYPE} == 2 ? 1 : 2)$</p> <p>Enabled: 0</p> <p>Parameter Name: <code>DSI_HOST_DSC_DEMUX_RAM_NINST</code></p>

Signal Descriptions

This chapter details all possible I/O signals in the IP. For configurable IP titles, your actual configuration might not contain all of these signals.

Inputs are on the left of the signal diagrams; outputs are on the right.

Attention: For configurable IP titles, do not use this document to determine the exact I/O footprint of the controller. It is for reference purposes only.

When you configure the controller in coreConsultant, you must access the I/O signals for your actual configuration at workspace/report/IO.html or workspace/report/IO.xml after you have completed the report creation activity. That report comes from the exact same source as this chapter but removes all the I/O signals that are not in your actual configuration. This does not apply to non-configurable IP titles. In addition, all parameter expressions are evaluated to actual values. Therefore, the widths might change depending on your actual configuration.

Some expressions might refer to TCL functions or procedures (sometimes identified as <functionof>) that coreConsultant uses to make calculations. The exact formula used by these TCL functions is not provided in this chapter. However, when you configure the controller in coreConsultant, all TCL functions and parameters are evaluated completely, and the resulting values are displayed where appropriate in the coreConsultant GUI reports.

In addition to describing the function of each signal, the signal descriptions in this chapter include the following information:

- **Active State:** Indicates whether the signal is active high or active low. When a signal is not intended to be used in a particular application, then this signal needs to be tied or driven to the inactive state (opposite of the active state).
- **Registered:** Indicates whether or not the signal is registered directly inside the IP boundary without intervening logic (excluding simple buffers). A value of No does not imply that the signal is not synchronous, only that there is some combinatorial logic between the signal's origin or destination register and the boundary of the controller. A value of N/A indicates that this information is not provided for this IP title.
- **Synchronous to:** Indicates which clocks in the IP sample this input (drive for an output). This clock might not be the same as the clock that your application logic should use to clock (sample/drive) this pin. For more details, consult the clock section in the databook.
- **Exists:** Name of configuration parameter that populates this signal in your configuration.

The I/O signals are grouped as follows:

- “[AMBA APB Signals](#)” on page [123](#)
- “[DSI Host Generic Signals](#)” on page [126](#)
- “[PHY PPI Interface Signals](#)” on page [127](#)
- “[DPI/eDPI Interface Signals](#)” on page [139](#)
- “[DPI Interface to 2-Port RAM Signals](#)” on page [144](#)
- “[DBI Interface Shared Signals](#)” on page [150](#)
- “[DBI Type A Clocked E Interface Signals](#)” on page [152](#)
- “[DBI Type B Interface Signals](#)” on page [153](#)
- “[DBI Write Interface to 2-Port RAM Signals](#)” on page [154](#)
- “[DBI Read Interface to 2-Port RAM Signals](#)” on page [156](#)
- “[Generic Write Interface to 2-Port RAM Signals](#)” on page [158](#)
- “[Generic Read Interface to 2-Port RAM Signals](#)” on page [161](#)
- “[DSI Host Generic Signals](#)” on page [164](#)
- “[Vendor Specific Interface Signals](#)” on page [165](#)
- “[PHY External Interface Signals](#)” on page [167](#)
- “[DSC Encoder Memory Interface Signals](#)” on page [169](#)

4.1 AMBA APB Signals

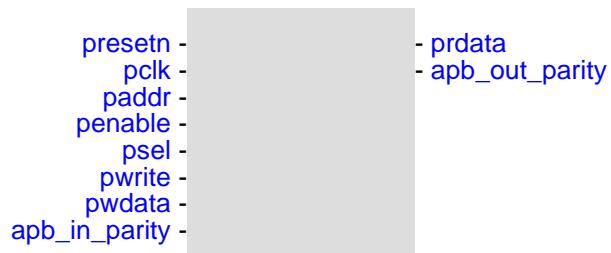


Table 4-1 AMBA APB Signals

Port Name	I/O	Description
presetn	I	<p>APB Asynchronous Reset Signal. This signal acts as a global reset. Although asynchronous, it must be maintained active for at least 1 pclk clock cycle.</p> <p>Exists: Always Synchronous To: Asynchronous Registered: N/A Power Domain: SINGLE_DOMAIN Active State: Low</p>
pclk	I	<p>APB clock Signal.</p> <p>Exists: Always Synchronous To: None Registered: N/A Power Domain: SINGLE_DOMAIN Active State: N/A</p>
paddr[9:0]	I	<p>APB Address bus.</p> <p>Exists: Always Synchronous To: pclk Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A</p>
penable	I	<p>APB Enable Signal.</p> <p>Exists: Always Synchronous To: pclk Registered: No Power Domain: SINGLE_DOMAIN Active State: High</p>

Port Name	I/O	Description
psel	I	<p>APB Slave Selection Signal.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
pwrite	I	<p>APB Write Enable Signal.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
pwdata[31:0]	I	<p>APB Write Data Bus.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
prdata[31:0]	O	<p>APB Read Data Bus.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
apb_in_parity[5:0]	I	<p>Parity Data. It is an input signal along with the APB interface, which indicates the correctness for the parity of APB input interface signals.</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of pwdata[7:0] ■ Bit 1 corresponding to the parity of pwdata[15:8] ■ Bit 2 corresponding to the parity of pwdata[23:16] ■ Bit 3 corresponding to the parity of pwdata[31:24] ■ Bit 4 corresponding to the parity of paddr [7:0] ■ Bit 5 corresponding to the parity of paddr [9:8], psel, penable and pwrite. <p>Exists: 0</p> <p>Synchronous To: pclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
apb_out_parity[3:0]	O	<p>Parity Data. It is an output signal along with the APB interface, which indicates the correctness for the parity prdata.</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of prdata[7:0] ■ Bit 1 corresponding to the parity of prdata[15:8] ■ Bit 2 corresponding to the parity of prdata[23:16] ■ Bit 3 corresponding to the parity of prdata[31:24] <p>Exists: 0 Synchronous To: pclk Registered: N/A Power Domain: SINGLE_DOMAIN Active State: N/A</p>

4.2 DSI Host Generic Signals



Table 4-2 DSI Host Generic Signals

Port Name	I/O	Description
interrupt	O	<p>Interruption Signal.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
diag_interrupt	O	<p>Diagnostic Interrupt.</p> <p>Exists: 0</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
scanmode	I	<p>Scan Mode Activation Signal.</p> <p>Exists: Always</p> <p>Synchronous To: Asynchronous</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

4.3 PHY PPI Interface Signals

rxclkesc	- txskewcalhs
lanebyteclk	- txblaneclk
phylock	- txckesc
phystopstateclklane	- physhutdownz
phyulpselectivenotclk	- phyrstz
phydirection	- phyforcepll
phystopstate0lane	- enableclk
I0txreadyhs	- txrequestclkhs
ulpsactivenot0lane	- txulpsclk
phystopstate1lane	- txulpsclkexit
ulpsactivenot1lane	- phyturnrequest
phystopstate2lane	- phyenable0lane
ulpsactivenot2lane	- txulpsesc0lane
phystopstate3lane	- txulpsexit0lane
ulpsactivenot3lane	- I0txrequestdatahs
txreadyesc	- I0txdatahs
rxdataesc	- I0txsendsynchs
rxlpdesc	- phyenable1lane
rxvalidesc	- txrequestesc1lane
rxtriggeresc	- txulpsesc1lane
rxulpsesc0lane	- txulpsexit1lane
phyerr0lane	- I1txrequestdatahs
	- I1xdatahs
	- I1txsendsynchs
	- phyenable2lane
	- txrequestesc2lane
	- txulpsesc2lane
	- txulpsexit2lane
	- I2txrequestdatahs
	- I2txdatahs
	- I2txsendsynchs
	- phyenable3lane
	- txrequestesc3lane
	- txulpsesc3lane
	- txulpsexit3lane
	- I3txrequestdatahs
	- I3txdatahs
	- txrequestesc
	- txlpdesc
	- txtriggeresc
	- txdataesc
	- txvalidesc

Table 4-3 PHY PPI Interface Signals

Port Name	I/O	Description
txskewcalhs	O	<p>High-Speed Transmit Skew Calibration.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
rxclkescl	I	<p>Lane 0 low-power RX Escape Mode Clock Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lanebyteclk	I	<p>PHY Lane Byte Clock</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
txblaneclk	O	<p>PHY Clock Lane Byte Return Signal Used for TX. This is a duplicate of lanebyteclk.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
txclkescl	O	<p>Lane 0 Low-Power TX Escape Mode Clock Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
physhutdownz	O	<p>PHY Digital and Analog Shut Down.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: pclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>
phyrstz	O	<p>PHY Digital Shutdown.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>

Port Name	I/O	Description
phylock	I	<p>PHY PLL Lock Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phyforcepll	O	<p>PHY Force PLL On.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: pclk</p> <p>Registered: ((DSI_HOST_COMBO == 1) (DSI_HOST_COMBO == 0 && DSI_HOST_DATAINTERFACE == 4)) ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
enableclk	O	<p>PHY Enable Clock Lane Generation.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: pclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txrequestclkhs	O	<p>PHY Request to Transmit High-Speed Clock.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsclk	O	<p>PHY Clock Lane Transmit ULPM.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsclkexit	O	<p>PHY Clock Lane Transmit Exit from ULPM.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
phystopstateclklane	I	<p>PHY Clock Lane Stop State Notification.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phyulpsactivenotclk	I	<p>This signal indicates that the clock lane is in the ULPM.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: Asynchronous</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>
phyturnrequest	O	<p>PHY Turn Request Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phydirection	I	<p>PHY Current Direction of Lane 0 Interconnection:</p> <ul style="list-style-type: none"> ■ 0 : TX ■ 1 : RX <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phyenable0lane	O	<p>PHY Lane 0 Enable Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsesc0lane	O	<p>PHY Data Lane 0 Transmit/Enter ULPM.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
txulpsexit0lane	O	<p>PHY Data Lane 0 Exit ULPM.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phystopstate0lane	I	<p>PHY Lane 0 is in Stop State.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I0txrequestdatahs	O	<p>Lane 0 TX High-Speed Transmit Data Sending Request.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I0txdatahs[(DSI_HOST_PPI_WIDTH-1):0]	O	<p>Lane 0 TX High-Speed Transmit Data Output.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Yes</p>
I0txreadyhs	I	<p>Lane 0 TX High-Speed Transmit Ready.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
ulpsactivenot0lane	I	<p>Lane 0 ULPM Active.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>

Port Name	I/O	Description
l0txsendsynchs	O	<p>C-PHY Lane 0 High-Speed Command to Transmit Sync Word.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_PHY == 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phyenable1lane	O	<p>PHY Lane 1 Enable Signal.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txrequestesc1lane	O	<p>PHY Data Lane 1 Request Transmit Escape Mode.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsesc1lane	O	<p>PHY Data Lane 1 Transmit/Enter ULPM.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsexit1lane	O	<p>PHY Data Lane 1 Exit ULPM.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phystopstate1lane	I	<p>PHY Lane 1 is in Stop State.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
I1txrequestdatahs	O	<p>Lane 1 TX High-Speed Transmit Data Sending Request.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I1txdatahs[(DSI_HOST_PPI_WIDTH-1):0]	O	<p>Lane 1 TX High-Speed Transmit Data Output.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
ulpsactivenot1lane	I	<p>Lane 1 ULPM Active.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2))</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>
I1txsendsynchs	O	<p>C-PHY Lane 1 High-Speed Command to Transmit Sync Word.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 2)) && ((DSI_HOST_PHY == 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phyenable2lane	O	<p>PHY Lane 2 Enable Signal.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
txrequestesc2lane	O	<p>PHY Data Lane 2 Request Transmit Escape Mode.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpesc2lane	O	<p>PHY Data Lane 2 Transmit/Enter ULPM.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsexit2lane	O	<p>PHY Data Lane 2 Exit ULPM.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phystopstate2lane	I	<p>PHY Lane 2 is in Stop State.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I2txrequestdatahs	O	<p>Lane 2 TX High-Speed Transmit Data Sending Request.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I2txdatahs[(DSI_HOST_PPI_WIDTH-1):0]	O	<p>Lane 2 TX High-Speed Transmit Data Output.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
ulpsactivenot2lane	I	<p>Lane 2 ULPM Active.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3))</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>
l2txsendsynchs	O	<p>C-PHY Lane 2 High-Speed Command to Transmit Sync Word.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 3)) && ((DSI_HOST_PHY == 2))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phyenable3lane	O	<p>PHY Lane 3 Enable Signal</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txrequestesc3lane	O	<p>PHY Data Lane 3 Request Transmit Escape Mode.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txulpsesc3lane	O	<p>PHY Data Lane 3 Transmit/Enter ULPM.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
txulpsexit3lane	O	<p>PHY Data Lane 3 Exit ULPM.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
phystopstate3lane	I	<p>PHY Lane 3 is in Stop State.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: Asynchronous</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I3txrequestdatahs	O	<p>Lane 3 TX High-Speed Transmit Data Sending Request.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
I3txdatahs[7:0]	O	<p>Lane 3 TX High-Speed Transmit Data Output.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
ulpsactivenot3lane	I	<p>Lane 3 ULPM Active.</p> <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_NUMBER_OF_LANES >= 4))</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>
txrequestesc	O	<p>Lane 0 Low-Power TX Escape Mode Transmit Request.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: txclkesc</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
txlpdtesc	O	<p>Lane 0 Low-Power TX Escape Mode Transmit Low-Power Data Request.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: txclkesc</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txtriggeresc[3:0]	O	<p>Lane 0 Low-Power TX Escape Mode Transmission Triggers.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: txclkesc</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txdataesc[7:0]	O	<p>Lane 0 Low-Power TX Escape Mode Low-Power Data Transmission.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: txclkesc</p> <p>Registered: DSI_HOST_COMBO ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
txvalidesc	O	<p>Lane 0 Low-Power TX Escape Mode Low-Power Data Valid Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: txclkesc</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
txreadyesc	I	<p>Lane 0 Low-Power TX Escape Mode Low-Power Data Transmit Ready Signal.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: txclkesc</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
rxdataesc[7:0]	I	<p>Lane 0 Low-Power RX Escape Mode Received Data.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: rxclkesc</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
rxlpdtesc	I	Lane 0 Low-Power RX Escape Mode Low-Power Data Receive Mode. Exists: !DSI_HOST_SNPS_PHY Synchronous To: rxclkesc Registered: No Power Domain: SINGLE_DOMAIN Active State: High
rxvalidesc	I	Lane 0 Low-Power RX Escape Mode Valid Receive Data. Exists: !DSI_HOST_SNPS_PHY Synchronous To: rxclkesc Registered: No Power Domain: SINGLE_DOMAIN Active State: High
rxtriggeresc[3:0]	I	Lane 0 Low-Power RX Escape Mode Received Triggers. Exists: !DSI_HOST_SNPS_PHY Synchronous To: (DSI_HOST_DBINTERFACE) (DSI_HOST_EDPIINTERFACE) ? "rxclkesc" : "None" Registered: No Power Domain: SINGLE_DOMAIN Active State: High
rxulpsesc0lane	I	Lane 0 ULPM Escape Code Receive. Exists: !DSI_HOST_SNPS_PHY Synchronous To: pclk Registered: Yes Power Domain: SINGLE_DOMAIN Active State: High
phyerr0lane[4:0]	I	PHY Lane 0 Error Input Signals. <ul style="list-style-type: none"> ■ Bit 0: ErrEsc escape entry error from Lane 0 ■ Bit 1: ErrSyncEsc low-power data transmission synchronization error from Lane 0 ■ Bit 2: Control error ErrControl from Lane 0 ■ Bit 3: LP0 contention error ErrContentionLP0 from Lane 0 ■ Bit 4: LP1 contention ErrContentionLP1 from Lane 0 Exists: !DSI_HOST_SNPS_PHY Synchronous To: lanebyteclk Registered: Yes Power Domain: SINGLE_DOMAIN Active State: High

4.4 DPI/eDPI Interface Signals

dipiclk	-
dipiclk_dsc	-
dpivsync_edpiwms	-
dpivsync	-
dpihsync	-
dipiixdata	-
dipiixdata2	-
dpiodataen	-
dpishutdown	-
dpicolorm	-
dpiupdatecfg	-
tear_request	-
edpi_in_parity	-

Table 4-4 DPI/eDPI Interface Signals

Port Name	I/O	Description
dipiclk	I	<p>Input Pixel Clock Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipiclk_dsc	I	<p>DSC Encoder clock. This clock must be synchronous with dpiclk. The frequency value should be equal to dpiclk / 2 for dual dsc mode and equal to dpiclk for single dsc mode.</p> <p>Exists: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dpivsync_edpiwms	I	<p>Write Memory Start Signal.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && ((DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpiclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
dpivsync	I	<p>Vertical Synchronism Signal.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (!!(DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Programmable</p>
dpihsync	I	<p>Horizontal Synchronism Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Programmable</p>
dipiidata[(DSI_HOST_PIXELDATAWIDTH-1):0]	I	<p>Video data. Video data is delivered one pixel per clock cycle.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpipclk</p> <p>Registered: DSI_HOST_DPIINTERFACE && DSI_HOST_DSC_ENC ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipiidata2[(DSI_HOST_PIXELDATAWIDTH-1):0]	I	<p>Video data for the second DSC encoder. Video data is delivered one pixel per clock cycle.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC_DPORT)</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
dodataen	I	<p>Video Data Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: (DSI_HOST_DPIINTERFACE && DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK) ? "dipiclk,dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Programmable</p>
dpishutdn	I	<p>Control Signal. It is used to shutdown the display.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpiclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Programmable</p>
dpicolorm	I	<p>Control Signal. It is used to switch between normal color and reduced color mode.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpiclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Programmable</p>
dpiupdatecfg	I	<p>Control Signal. This signal is used to indicate that the next frame will have new video configuration.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpiclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
edpihalt	O	<p>Halt Indication on Video Interface.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
tear_request	I	<p>Activate Tearing Effect by Hardware.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && ((DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpipclk</p> <p>Registered: (DSI_HOST_DPIINTERFACE == 4) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
edpite	O	<p>Tearing Effect Indication.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && ((DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
edpi_in_parity[4:0]	I	<p>Parity Data. It is an input signal along with the eDPI interface, which indicates the correctness for the parity of eDPI input interface signals including dpipixdata:</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of dpipixdata[7:0] ■ Bit 1 corresponding to the parity of dpipixdata[15:8] ■ Bit 2 corresponding to the parity of dpipixdata[23:16] ■ Bit 3 corresponding to the parity of dpiwsync_ed-pihsync,dpihdata[29:24] ■ Bit 4 corresponding to the parity of tear_request,dpiupdatecfg,dpicolorm,dpisutdn,dpidataen <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (0)</p> <p>Synchronous To: dpipclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
edpi_out_parity	O	<p>Parity Data. It is an output signal along with the eDPI interface, which indicates the correctness for the parity of eDPI output interface signals:</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of edpihalt, edpите <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (0)</p> <p>Synchronous To: dpipclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

4.5 DPI Interface to 2-Port RAM Signals

dpipixelmemin -
 dpipixelmemin_2enc -

 - dpipixelwclk
 - dpipixelerclk
 - dpipixelmemout
 - dpipixelrdaddr
 - dpipixelwraddr
 - dpipixelwrenable
 - dpipixelrdenable
 - dpipixelmemselect
 - dpipixelwclk_2enc
 - dpipixelerclk_2enc
 - dpipixelmemout_2enc
 - dpipixelrdaddr_2enc
 - dpipixelwraddr_2enc
 - dpipixelwrenable_2enc
 - dpipixelrdenable_2enc
 - dpipixelmemselect_2enc
 - dpiraddrparity
 - dpiwaddrparity

Table 4-5 **DPI Interface to 2-Port RAM Signals**

Port Name	I/O	Description
dpipixelwclk	O	<p>Memory Write Clock Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dpipixelerclk	O	<p>Memory Read Clock Signal</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
dipipixelmemin[(DSI_HOST_DPI_PIXEL_RAM_DWIDTH-1):0]	I	<p>Memory Data Read Bus (extended with ECC 7-bit when using AP). This signal outputs the data pointed by address, when read enable is active.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: (DSI_HOST_DPIINTERFACE && (DSI_HOST_DUAL_DSC_ENC && DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM) ((DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM == 0) && (DSI_HOST_SNPS_SYNC_RD_FIFOS == 0))) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipipixelmemout[(DSI_HOST_DPI_PIXEL_RAM_DWIDTH-1):0]	O	<p>Memory Data Write Bus (extended with ECC 7-bit when using AP).</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: (DSI_HOST_DPIINTERFACE && DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK) ? "dipiclk,dipiclk_dsc" : "dipiclk"</p> <p>Registered: (((DSI_HOST_DSC_ENC==0) && (SNPS_RSVDPARAM_1==0) && (DSI_HOST_64_DATA_INT==1)) ((DSI_HOST_64_DATA_INT==0) && (DSI_HOST_EDPIINTERFACE==0))) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipipixelrdaddr[(DSI_HOST_DPI_RB_RAM_ADDRWIDTH-1):0]	O	<p>Memory Read Address Bus. The memory size is parameterizable.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: (DSI_HOST_DPIINTERFACE && (DSI_HOST_DSC_ENC_TYPE!=2)) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
dipipixelwraddr[(DSI_HOST_DPI_RB_RA M_ADDRWIDTH-1):0]	O	<p>Memory Write Address Bus. The memory size is parameterizable.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: (DSI_HOST_DPIINTERFACE && (DSI_HOST_DSC_ENC_TYPE!=2)) ? "dipiclk,dipiclk_dsc" : "dipiclk"</p> <p>Registered: (DSI_HOST_DPIINTERFACE && (DSI_HOST_DSC_ENC_TYPE!=2)) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipipixelwenable	O	<p>Memory Write Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: (DSI_HOST_DPIINTERFACE && (DSI_HOST_DSC_ENC_TYPE!=2)) ? "dipiclk,dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dipipixelrdenable	O	<p>Memory Read Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dipipixelmemselect	O	<p>Memory Interface Select.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
dipipixelwclk_2enc	O	<p>Dual DSC Encoder - 2nd Encoder Memory Write Clock Signal.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipipixelrclk_2enc	O	<p>Dual DSC Encoder - 2nd Encoder Memory Read Clock Signal</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipipixelmemin_2enc[(DSI_HOST_DPI_RB_RAM_DATAWIDTH-1):0]	I	<p>Dual DSC Encoder - 2nd Encoder Memory Data Read Bus.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dipipixelmemout_2enc[(DSI_HOST_DPI_RB_RAM_DATAWIDTH-1):0]	O	<p>Dual DSC Encoder - 2nd Encoder Memory Data Write Bus.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: dipiclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
dpipixelrdaddr_2enc[(DSI_HOST_RB_SE_C_DSC_RAM_ADDRWIDTH-1):0]	O	<p>Dual DSC Encoder - 2nd Encoder Memory Read Address Bus. The memory size is parameterizable.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dpipixelwraddr_2enc[(DSI_HOST_RB_SE_C_DSC_RAM_ADDRWIDTH-1):0]	O	<p>Dual DSC Encoder - 2nd Encoder Memory Write Address Bus. The memory size is parameterizable.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: dpiclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dpipixelwenable_2enc	O	<p>Dual DSC Encoder - 2nd Encoder Memory Write Enable Signal.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: dpiclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dpipixelrdenable_2enc	O	<p>Dual DSC Encoder - 2nd Encoder Memory Read Enable Signal.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
dipipixelmemselect_2enc	O	<p>Dual DSC Encoder - 2nd Encoder Memory Interface Select.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))) && (DSI_HOST_DUAL_DSC_ENC) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dpiraddrparity[1:0]	O	<p>Parity Data. DPI Memory Read operation Parity Bus. It is the parity of dpiraddr and dpirdenable:</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of dpiraddr[6:0], dpirdenable ■ Bit 1 corresponding to the parity of dpiraddr[11:7] <p>Exists: 0</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dpiwaddrparity[1:0]	O	<p>Parity Data.. DPI Memory Write Operation Parity Bus. It is the parity of dpiwraddr[11:0] and dpiwrenable.</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of dpiwraddr [6:0], dpiwrenable ■ Bit 1 corresponding to the parity of dpiwraddr [11:7] <p>Exists: 0</p> <p>Synchronous To: dpiclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

4.6 DBI Interface Shared Signals

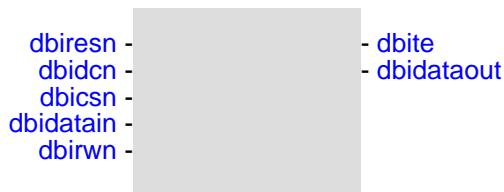


Table 4-6 DBI Interface Shared Signals

Port Name	I/O	Description
dbiresn	I	<p>DBI Reset Signal Type A Fixed E.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: Low</p>
dbidcn	I	<p>DBI Data/Command Indicates data when signal is high. Indicates command when signal is low.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: DSI_HOST_DBINTERFACE_TYPE == 0 ? "dbicsn" : (DSI_HOST_DBINTERFACE_TYPE == 1 ? "dbie" : "dbirwrck & dbirdck")</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: -Type A Fixed E: High; -Type A Clocked E and Type B: N/A</p>
dbicsn	I	<p>DBI Chip Select Type A Clocked E</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: DSI_HOST_DBINTERFACE_TYPE == 0 ? "dbicsn" : (DSI_HOST_DBINTERFACE_TYPE == 1 ? "dbie" : "dbirwrck")</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: -Type A Clocked E and Type B: Low; -Type A Fixed E: N/A.</p>

Port Name	I/O	Description
dbite	O	<p>DBI Tear Effect</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dbidatain[15:0]	I	<p>DBI Data Input Bus</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: DSI_HOST_DBINTERFACE_TYPE == 0 ? "dbicsn" : (DSI_HOST_DBINTERFACE_TYPE == 1 ? "dbie" : "dbirdck")</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbidataout[15:0]	O	<p>DBI Data Output Bus</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: DSI_HOST_DBINTERFACE_TYPE == 0 ? "dbicsn" : (DSI_HOST_DBINTERFACE_TYPE == 1 ? "dbie" : "dbirdck")</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbirwn	I	<p>DBI Read/Write. For Type A Fixed E and Type A Clocked E the DWC_mipi_dsi_host writes the data at the rising edge or reads at the falling edge for Type A Fixed E and Type A Clocked E.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))) && (!((DSI_HOST_DBINTERFACE_TYPE == 2) && (DSI_HOST_DBINTERFACE)))</p> <p>Synchronous To: DSI_HOST_DBINTERFACE_TYPE == 0 ? "dbicsn" : "dbie"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: -Type A Fixed E: High; -Type A Clocked E: N/A.</p>

4.7 DBI Type A Clocked E Interface Signals

dbie -

Table 4-7 DBI Type A Clocked E Interface Signals

Port Name	I/O	Description
dbie	I	<p>DBI E Clock. The DWC_mipi_dsi_host reads the data at the falling edge or writes at the rising edge.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))) && (((DSI_HOST_DBINTERFACE_TYPE == 1) && (DSI_HOST_DBINTERFACE)))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

4.8 DBI Type B Interface Signals

dbirwrck -
dbirdck -

Table 4-8 DBI Type B Interface Signals

Port Name	I/O	Description
dbirwrck	I	<p>DBI Write for Type B. The DWC_mipi_dsi_host reads the information at the rising edge for DBI Type B interface.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))) && (((DSI_HOST_DBINTERFACE_TYPE == 2) && (DSI_HOST_DBINTERFACE)))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbirdck	I	<p>DBI Read. The DWC_mipi_dsi_host writes the information at the falling edge.</p> <p>Exists: (((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))) && (((DSI_HOST_DBINTERFACE_TYPE == 2) && (DSI_HOST_DBINTERFACE)))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

4.9 DBI Write Interface to 2-Port RAM Signals

dbipldmemin -

- dbipldwclk
- dbipldrclk
- dbipldmemout
- dbiplrdaddr
- dbiplwraddr
- dbipldwrenable
- dbiplrdenable
- dbipldmemselect

Table 4-9 DBI Write Interface to 2-Port RAM Signals

Port Name	I/O	Description
dbipldwclk	O	<p>DBI PLD WR Memory Write Clock Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbipldrclk	O	<p>DBI PLD WR Memory Read Clock Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbipldmemin[(DSI_HOST_DBIPLDDATAW IDTH-1):0]	I	<p>DBI PLD WR Memory Data Read Bus. This signal outputs the data pointed by the address, when read enable is active.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_SNPS_SYNC_RD_FIFOS ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbipldmemout[(DSI_HOST_DBIPLDDATA WIDTH-1):0]	O	<p>DBI PLD WR Memory Data Write Bus.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_COMBO ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
dbipldrdaddr[(DSI_HOST_DBIPLDADDR WIDTH-1):0]	O	<p>DBI PLD WR Memory Read Address Bus. Configurable RAM size.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbipldwraaddr[(DSI_HOST_DBIPLDADDR WIDTH-1):0]	O	<p>DBI PLD WR Memory Write Address Bus. Configurable RAM size.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbipldwrenable	O	<p>DBI PLD WR Memory Write Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dbipldrdenable	O	<p>DBI PLD WR Memory Read Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dbipldmemselect	O	<p>DBI PLD WR Memory Select</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

4.10 DBI Read Interface to 2-Port RAM Signals

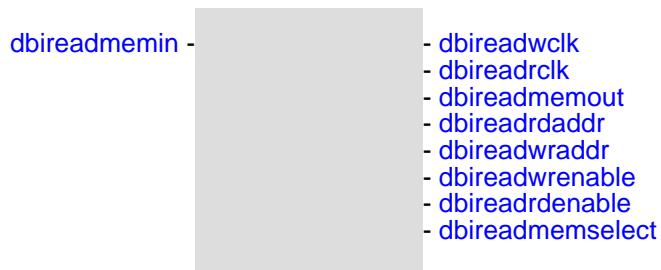


Table 4-10 DBI Read Interface to 2-Port RAM Signals

Port Name	I/O	Description
dbireadwclk	O	<p>DBI PLD RD Memory Write Clock Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbireadrclk	O	<p>DBI PLD RD Memory Read Clock Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbireadmemin[7:0]	I	<p>DBI PLD RD Memory Data Read Bus. This signal outputs the data pointed by the address, when read enable is active.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: DSI_HOST_SNPS_SYNC_RD_FIFOS ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbireadmemout[7:0]	O	<p>DBI PLD RD Memory Data Write Bus.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
dbireaddraddr[(DSI_HOST_DBIREADPLD ADDRWIDTH-1):0]	O	<p>DBI PLD RD Memory Read Address Bus. Configurable RAM size.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbireadwraddr[(DSI_HOST_DBIREADPL DADDRWIDTH-1):0]	O	<p>DBI PLD RD Memory Write Address Bus. Configurable RAM size.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
dbireadwenable	O	<p>DBI PLD RD Memory Write Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dbireadrdenable	O	<p>DBI PLD RD Memory Read Enable Signal.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
dbireadmemselect	O	<p>DBI PLD RD Memory Select.</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

4.11 Generic Write Interface to 2-Port RAM Signals



Table 4-11 Generic Write Interface to 2-Port RAM Signals

Port Name	I/O	Description
genericpldwclk	O	Generic PLD WR Memory Write Clock Signal. Exists: Always Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A
genericpldrclk	O	Generic PLD WR Memory Read Clock Signal Exists: Always Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A
genericpldmemin[(DSI_HOST_GEN_PLD _RAM_DWIDTH-1):0]	I	Generic PLD WR Memory Data Read Bus (extended with ECC 7-bit when using AP). This signal outputs the data pointed by address when read enable is active. Exists: Always Synchronous To: lanebyteclk Registered: DSI_HOST_SNPS_SYNC_RD_FIFOS ? "No" : "Yes" Power Domain: SINGLE_DOMAIN Active State: N/A
genericpldmemout[(DSI_HOST_GEN_PL D_RAM_DWIDTH-1):0]	O	Generic PLD WR Memory Data Write Bus (extended with ECC 7-bit when using AP). Exists: Always Synchronous To: pclk Registered: Yes Power Domain: SINGLE_DOMAIN Active State: N/A

Port Name	I/O	Description
genericplrdaddr[(DSI_HOST_GENERICP LDADDRWIDTH-1):0]	O	<p>Generic PLD WR Memory Read Address Bus. Configurable RAM size.</p> <p>Exists: Always</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
genericpldwraddr[(DSI_HOST_GENERIC PLDADDRWIDTH-1):0]	O	<p>Generic PLD WR Memory Write Address Bus. Configurable RAM size.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
genericpldwrenable	O	<p>Generic PLD WR Memory Write Enable Signal.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
genericplrdenable	O	<p>Generic PLD WR Memory Read Enable Signal.</p> <p>Exists: Always</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
genericpldmemselect	O	<p>Generic PLD WR Memory Select Signal.</p> <p>Exists: Always</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
genpldraddrparity	O	<p>Parity Data. Generic Write Memory Read operation Parity Bus. It is the parity of genericplrdaddr[5:0] and genericplrdenable.</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of genericplrdaddr[5:0], genericplrdenable <p>Exists: 0</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
genpldaddrparity	O	<p>Parity Data. Generic Write Memory Write Operation Parity Bus. It is the parity of genericpldwraddr[5:0] and genericpldwrenable.</p> <ul style="list-style-type: none">■ Bit 0 corresponding to the parity of genericpldwraddr[5:0],genericpldwrenable <p>Exists: 0 Synchronous To: pclk Registered: N/A Power Domain: SINGLE_DOMAIN Active State: N/A</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

4.12 Generic Read Interface to 2-Port RAM Signals

genericreadmemin -	<ul style="list-style-type: none"> - genericreadwclk - genericreadrclk - genericreadmemout - genericreadrdaddr - genericreadwraddr - genericreadwrenable - genericreadrdenable - genericreadmemselect - genreadraddrparity - genreadwaddrparity
--------------------	---

Table 4-12 Generic Read Interface to 2-Port RAM Signals

Port Name	I/O	Description
genericreadwclk	O	Generic PLD RD Memory Write Clock Signal. Exists: Always Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A
genericreadrclk	O	Generic PLD RD Memory Read Clock Signal. Exists: Always Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A
genericreadmemin[(DSI_HOST_GEN_RDAD_RAM_DWIDTH-1):0]	I	Generic PLD RD Memory Data Read Bus (extended with ECC 7-bit when using AP). Exists: Always Synchronous To: pcik Registered: DSI_HOST_SNPS_SYNC_RD_FIFOS ? "No" : "Yes" Power Domain: SINGLE_DOMAIN Active State: N/A
genericreadmemout[(DSI_HOST_GEN_RDAD_RAM_DWIDTH-1):0]	O	Generic PLD RD Memory Data Write Bus (extended with ECC 7-bit when using AP). Exists: Always Synchronous To: lanebyteclk Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A

Port Name	I/O	Description
genericreadrdaddr[(DSI_HOST_GENREA_DPLDADDRWIDTH-1):0]	O	<p>Generic PLD RD Memory Read Address Bus. Configurable RAM size.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
genericreadwraddr[(DSI_HOST_GENREA_DPLDADDRWIDTH-1):0]	O	<p>Generic PLD RD Memory Write Address Bus. Configurable RAM size.</p> <p>Exists: Always</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
genericreadwenable	O	<p>Generic PLD RD Memory Write Enable Signal.</p> <p>Exists: Always</p> <p>Synchronous To: lanebyteclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
genericreadrdenable	O	<p>Generic PLD RD Memory Read Enable Signal.</p> <p>Exists: Always</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
genericreadmemselect	O	<p>Generic PLD RD Memory Select.</p> <p>Exists: Always</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
genreadraddrparity	O	<p>Parity Data. Generic Read Memory Read operation Parity Bus. It is the parity of genericreadrdaddr[4:0] and genericreadrdenable.</p> <ul style="list-style-type: none"> ■ Bit 0 corresponding to the parity of genericreadrdaddr[4:0], genericreadrdenable <p>Exists: 0</p> <p>Synchronous To: pclk</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
genreadwaddrparity	O	<p>Parity Data. Generic Read Memory Write Operation Parity Bus. It is the parity of genericreadwraddr[4:0] and genericreadwrenable.</p> <ul style="list-style-type: none">■ Bit 0 corresponding to the parity of genericreadwraddr[4:0], generic-readwrenable <p>Exists: 0 Synchronous To: lanebyteclk Registered: N/A Power Domain: SINGLE_DOMAIN Active State: N/A</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

4.13 DSI Host Generic Signals

- phy_mode

Table 4-13 DSI Host Generic Signals

Port Name	I/O	Description
phy_mode	O	<p>PHY Select pin.</p> <ul style="list-style-type: none">■ 0: D-PHY mode■ 1: C-PHY mode <p>Exists: (!DSI_HOST_SNPS_PHY) && ((DSI_HOST_PHY == 2))</p> <p>Synchronous To: None</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

4.14 Vendor Specific Interface Signals

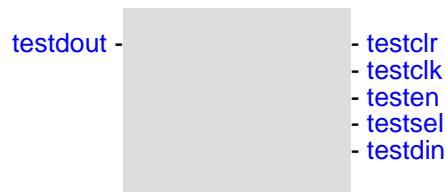


Table 4-14 Vendor Specific Interface Signals

Port Name	I/O	Description
testclr	O	<p>PHY Test Clear Signal. Exists: !DSI_HOST_SNPS_PHY Synchronous To: pclk Registered: DSI_HOST_COMBO ? "No" : "Yes" Power Domain: SINGLE_DOMAIN Active State: High</p>
testclk	O	<p>PHY Test Clock Signal. Exists: !DSI_HOST_SNPS_PHY Synchronous To: pclk Registered: DSI_HOST_COMBO ? "No" : "Yes" Power Domain: SINGLE_DOMAIN Active State: N/A</p>
testen	O	<p>PHY Test Enable. Exists: !DSI_HOST_SNPS_PHY Synchronous To: pclk Registered: DSI_HOST_COMBO ? "No" : "Yes" Power Domain: SINGLE_DOMAIN Active State: High</p>
testsel	O	<p>PHY BiDIR MUX selector. Exists: !DSI_HOST_SNPS_PHY Synchronous To: pclk Registered: Yes Power Domain: SINGLE_DOMAIN Active State: High</p>
testdin[7:0]	O	<p>PHY Test Data Output Port. Exists: !DSI_HOST_SNPS_PHY Synchronous To: pclk Registered: DSI_HOST_COMBO ? "No" : "Yes" Power Domain: SINGLE_DOMAIN Active State: N/A</p>

Port Name	I/O	Description
testdout[7:0]	I	<p>PHY Test Data Input Port.</p> <p>Exists: !DSI_HOST_SNPS_PHY</p> <p>Synchronous To: pclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

4.15 PHY External Interface Signals

cfg_clk -
 scanbyteclk -
 scantxclkes -
 scanrxclkes -
 refclk -

Table 4-15 PHY External Interface Signals

Port Name	I/O	Description
cfg_clk	I	<p>PHY configuration clock used for the initialization of the PHY. It is also used for exiting ULPS state.</p> <p>Exists: DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
scanbyteclk	I	<p>PHY scan clock used for lanebyteclk.</p> <p>Exists: DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
scantxclkes	I	<p>PHY scan clock used for txclkes.</p> <p>Exists: DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
scanrxclkes	I	<p>PHY scan clock used for rxclkes.</p> <p>Exists: DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
refclk	I	<p>PHY reference clock.</p> <p>Exists: DSI_HOST_SNPS_PHY</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

4.16 DSC Encoder Memory Interface Signals

demux_ram_datain_2enc -
demux_ram_datain -
balance_ram_rd_data -
balance_ram_rd_data_2enc -
lb_ram_rd_data -
lb_ram_rd_data_2enc -
se_size_ram_rd_data -
se_size_ram_rd_data_2enc -

- demux_ram_clk_2enc
- demux_ram_dataout_2enc
- demux_ram_readaddress_2enc
- demux_ram_writeaddress_2enc
- demux_ram_writeena_2enc
- demux_ram_select_2enc
- demux_ram_readena_2enc
- demux_ram_clk
- demux_ram_dataout
- demux_ram_readaddress
- demux_ram_writeaddress
- demux_ram_writeena
- demux_ram_select
- demux_ram_readena
- balance_ram_clk
- balance_ram_rd_addr
- balance_ram_rd_enable
- balance_ram_select
- balance_ram_wr_addr
- balance_ram_wr_data
- balance_ram_wr_enable
- balance_ram_clk_2enc
- balance_ram_rd_addr_2enc
- balance_ram_rd_enable_2enc
- balance_ram_select_2enc
- balance_ram_wr_addr_2enc
- balance_ram_wr_data_2enc
- balance_ram_wr_enable_2enc
- lb_ram_clk
- lb_ram_rd_addr
- lb_ram_rd_enable
- lb_ram_select
- lb_ram_wr_addr
- lb_ram_wr_data
- lb_ram_wr_enable
- lb_ram_clk_2enc
- lb_ram_rd_addr_2enc
- lb_ram_rd_enable_2enc
- lb_ram_select_2enc
- lb_ram_wr_addr_2enc
- lb_ram_wr_data_2enc
- lb_ram_wr_enable_2enc
- se_size_ram_clk
- se_size_ram_rd_addr
- se_size_ram_rd_enable
- se_size_ram_select
- se_size_ram_wr_addr
- se_size_ram_wr_data
- se_size_ram_wr_enable
- se_size_ram_clk_2enc
- se_size_ram_rd_addr_2enc
- se_size_ram_rd_enable_2enc
- se_size_ram_select_2enc
- se_size_ram_wr_addr_2enc
- se_size_ram_wr_data_2enc
- se_size_ram_wr_enable_2enc

Table 4-16 DSC Encoder Memory Interface Signals

Port Name	I/O	Description
demux_ram_datain_2enc[(DSC_SLICE_D EMUX_RAM_DWIDTH-1):0]	I	<p>Second Demultiplexer RAM read data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: dpipclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_clk_2enc	O	<p>Second Demultiplexer RAM clock</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_dataout_2enc[(DSC_SLICE_ DEMUX_RAM_DWIDTH-1):0]	O	<p>Second Demultiplexer RAM write data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_readaddress_2enc[(DSC_SLI CE_DEMUX_RAM_AWIDTH-1):0]	O	<p>Second Demultiplexer RAM read address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: dpipclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_writeaddress_2enc[(DSC_SL ICE_DEMUX_RAM_AWIDTH-1):0]	O	<p>Second Demultiplexer RAM write address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: dpipclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
demux_ram_writeena_2enc	O	<p>Second Demultiplexer RAM write enable signal.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
demux_ram_select_2enc	O	<p>Second Demultiplexer RAM select signal.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
demux_ram_readena_2enc	O	<p>Second Demultiplexer RAM read enable.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX) && (!DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
demux_ram_datain[(DSC_SLICE_DEMUX _RAM_DWIDTH-1):0]	I	<p>Slice Demultiplexer RAM read data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: dpipclk</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_clk	O	<p>Slice Demultiplexer RAM clock.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
demux_ram_dataout[(DSC_SLICE_DEMU_X_RAM_DWIDTH-1):0]	O	<p>Slice Demultiplexer RAM write data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: dpipclk</p> <p>Registered: (DSI_HOST_DPIINTERFACE && DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_readaddress[(DSC_SLICE_DEMUX_RAM_AWIDTH-1):0]	O	<p>Slice Demultiplexer RAM read address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: dpipclk</p> <p>Registered: (DSI_HOST_DPIINTERFACE && DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM) ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_writeaddress[(DSC_SLICE_DEMUX_RAM_AWIDTH-1):0]	O	<p>Slice Demultiplexer RAM write address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: dpipclk</p> <p>Registered: (DSI_HOST_DPIINTERFACE && DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM) ? "No" : "Yes"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
demux_ram_writeena	O	<p>Slice Demultiplexer RAM write enable signal.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: dpipclk</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
demux_ram_select	O	<p>Slice Demultiplexer RAM select signal.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
demux_ram_readena	O	<p>Slice Demultiplexer RAM read enable.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC_INCDEMUX)</p> <p>Synchronous To: (DSI_HOST_DUAL_DSC_ENC_INCDEMUX & (DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM == 0)) ? "None" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
balance_ram_clk	O	<p>Encoder Balance RAM clock.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_rd_data[(DSC_BALANCE_RAM_COMP_DWIDTH-1):0]	I	<p>Encoder Balance RAM read data bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_rd_addr[(DSC_BALANCE_RAM_COMP_AWIDTH-1):0]	O	<p>Encoder Balance RAM read address.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: DSI_HOST_DPIINTERFACE ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_rd_enable[(DSC_NUM_COMPONENTS-1):0]	O	<p>Encoder Balance RAM read enable.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
balance_ram_select	O	<p>Encoder Balance RAM select.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
balance_ram_wr_addr[(DSC_BALANCE_RAM_COMP_AWIDTH-1):0]	O	<p>Encoder Balance RAM write address bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: DSI_HOST_DPIINTERFACE ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_wr_data[(DSC_BALANCE_RAM_COMP_DWIDTH-1):0]	O	<p>Encoder Balance RAM write data bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_wr_enable[(DSC_NUM_COMPONENTS-1):0]	O	<p>Encoder Balance RAM write enable.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
balance_ram_clk_2enc	O	<p>Second DSC Encoder Balance RAM clock.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_rd_data_2enc[(DSC_BALANCE_RAM_COMP_DWIDTH-1):0]	O	<p>Second DSC Encoder Balance RAM read data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
balance_ram_rd_addr_2enc[(DSC_BALANCE_RAM_COMP_AWIDTH-1):0]	O	<p>Second DSC Encoder Balance RAM read address,</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: (DSI_HOST_DUAL_DSC_ENC && DSI_HOST_DPIINTERFACE) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
balance_ram_rd_enable_2enc[(DSC_NUM_COMPONENTS-1):0]	O	<p>Second DSC Encoder Balance RAM read enable signal. Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk" Registered: No Power Domain: SINGLE_DOMAIN Active State: High</p>
balance_ram_select_2enc	O	<p>Second DSC Encoder Balance RAM select. Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: High</p>
balance_ram_wr_addr_2enc[(DSC_BALANCE_RAM_COMP_AWIDTH-1):0]	O	<p>Second DSC Encoder Balance RAM write address bus. Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk" Registered: (DSI_HOST_DUAL_DSC_ENC && DSI_HOST_DPIINTERFACE) ? "Yes" : "No" Power Domain: SINGLE_DOMAIN Active State: N/A</p>
balance_ram_wr_data_2enc[(DSC_BALANCE_RAM_COMP_DWIDTH-1):0]	O	<p>Second DSC Encoder Balance RAM write data bus. Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk" Registered: No Power Domain: SINGLE_DOMAIN Active State: N/A</p>
balance_ram_wr_enable_2enc[(DSC_NUM_COMPONENTS-1):0]	O	<p>Second DSC Encoder Balance RAM write enable bus. Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk" Registered: No Power Domain: SINGLE_DOMAIN Active State: High</p>
lb_ram_rd_data[(DSC_LB_RAM_DWIDTH-1):0]	I	<p>Line Buffer RAM read data bus. Exists: DSI_HOST_DSC_ENC Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk" Registered: Yes Power Domain: SINGLE_DOMAIN Active State: N/A</p>

Port Name	I/O	Description
lb_ram_clk	O	<p>Line Buffer RAM clock.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_rd_addr[(DSC_LB_RAM_AWIDTH -1):0]	O	<p>Line Buffer RAM read address bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dpipclk_dsc" : "dpipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_rd_enable	O	<p>Line Buffer RAM read enable.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dpipclk_dsc" : "dpipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
lb_ram_select	O	<p>Line Buffer RAM select.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
lb_ram_wr_addr[(DSC_LB_RAM_AWIDT H-1):0]	O	<p>Line Buffer RAM write address bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dpipclk_dsc" : "dpipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_wr_data[(DSC_LB_RAM_DWIDT H-1):0]	O	<p>Line Buffer RAM write data bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dpipclk_dsc" : "dpipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
lb_ram_wr_enable	O	<p>Line Buffer RAM write enable.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
lb_ram_rd_data_2enc[(DSC_LB_RAM_D WIDTH-1):0]	I	<p>Second DSC Encoder Line Buffer RAM read data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_clk_2enc	O	<p>Second DSC Encoder Line Buffer RAM clock.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_rd_addr_2enc[(DSC_LB_RAM_A WIDTH-1):0]	O	<p>Second DSC Encoder Line Buffer RAM read address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_rd_enable_2enc	O	<p>Second DSC Encoder Line Buffer RAM read enable.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
lb_ram_select_2enc	O	<p>Second DSC Encoder Line Buffer RAM select.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Port Name	I/O	Description
lb_ram_wr_addr_2enc[(DSC_LB_RAM_A WIDTH-1):0]	O	<p>Second DSC Encoder Line Buffer RAM write address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_wr_data_2enc[(DSC_LB_RAM_D WIDTH-1):0]	O	<p>Second DSC Encoder Line Buffer RAM write data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
lb_ram_wr_enable_2enc	O	<p>Second DSC Encoder Line Buffer RAM write enable.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
se_size_ram_clk	O	<p>Syntax Element Size RAM clock.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: None</p> <p>Registered: N/A</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
se_size_ram_rd_data[(DSC_SE_SIZE_R AM_DWIDTH-1):0]	O	<p>Syntax Element Size RAM read data bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
se_size_ram_rd_addr[(DSC_SE_SIZE_R AM_AWIDTH-1):0]	O	<p>Syntax Element Size RAM read address bus.</p> <p>Exists: DSI_HOST_DSC_ENC</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipclk_dsc" : "dipclk"</p> <p>Registered: DSI_HOST_DPIINTERFACE ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
se_size_ram_rd_enable	O	Syntax Element Size RAM read enable. Exists: DSI_HOST_DSC_ENC Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: High
se_size_ram_select	O	Syntax Element Size RAM select. Exists: DSI_HOST_DSC_ENC Synchronous To: None Registered: No Power Domain: SINGLE_DOMAIN Active State: High
se_size_ram_wr_addr[(DSC_SE_SIZE_RAM_AWIDTH-1):0]	O	Syntax Element Size RAM write address bus. Exists: DSI_HOST_DSC_ENC Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk" Registered: DSI_HOST_DPIINTERFACE ? "Yes" : "No" Power Domain: SINGLE_DOMAIN Active State: N/A
se_size_ram_wr_data[(DSC_SE_SIZE_RAM_DWIDTH-1):0]	O	Syntax Element Size RAM write data bus. Exists: DSI_HOST_DSC_ENC Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk" Registered: Yes Power Domain: SINGLE_DOMAIN Active State: N/A
se_size_ram_wr_enable	O	Syntax Element Size RAM write enable. Exists: DSI_HOST_DSC_ENC Synchronous To: DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK ? "dipiclk_dsc" : "dipiclk" Registered: No Power Domain: SINGLE_DOMAIN Active State: High
se_size_ram_clk_2enc	O	Second DSC Encoder Syntax Element RAM clock. Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC) Synchronous To: None Registered: N/A Power Domain: SINGLE_DOMAIN Active State: N/A

Port Name	I/O	Description
se_size_ram_rd_data_2enc[(DSC_SE_SIZE_RAM_DWIDTH-1):0]	I	<p>Second DSC Encoder Syntax Element RAM read data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
se_size_ram_rd_addr_2enc[(DSC_SE_SIZE_RAM_AWIDTH-1):0]	O	<p>Second DSC Encoder Syntax Element RAM read address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: (DSI_HOST_DPIINTERFACE && DSI_HOST_DUAL_DSC_ENC) ? "Yes" : "No"</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
se_size_ram_rd_enable_2enc	O	<p>Second DSC Encoder Syntax Element RAM read enable.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
se_size_ram_select_2enc	O	<p>Second DSC Encoder Syntax Element RAM select.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: None</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>
se_size_ram_wr_addr_2enc[(DSC_SE_SIZE_RAM_AWIDTH-1):0]	O	<p>Second DSC Encoder Syntax Element RAM write address bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>
se_size_ram_wr_data_2enc[(DSC_SE_SIZE_RAM_DWIDTH-1):0]	O	<p>Second DSC Encoder Syntax Element RAM write data bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dipiclk_dsc" : "dipiclk"</p> <p>Registered: Yes</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: N/A</p>

Port Name	I/O	Description
se_size_ram_wr_enable_2enc	O	<p>Second DSC Encoder Syntax Element RAM write enable bus.</p> <p>Exists: (DSI_HOST_DSC_ENC) && (DSI_HOST_DUAL_DSC_ENC)</p> <p>Synchronous To: DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX ? "dpipclk_dsc" : "dpipclk"</p> <p>Registered: No</p> <p>Power Domain: SINGLE_DOMAIN</p> <p>Active State: High</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

Register Descriptions

This chapter details all possible registers in the controller. They are arranged hierarchically into maps and blocks (banks). For configurable IP titles, your actual configuration might not contain all of these registers.

Attention: For configurable IP titles, do not use this document to determine the exact attributes of your register map. It is for reference purposes only.

When you configure the controller in coreConsultant, you must access the register attributes for your actual configuration at `workspace/report/ComponentRegisters.html` or `workspace/report/ComponentRegisters.xml` after you have completed the report creation activity. That report comes from the exact same source as this chapter but removes all the registers that are not in your actual configuration. This does not apply to non-configurable IP titles. In addition, all parameter expressions are evaluated to actual values. Therefore, the Offset and Memory Access values might change depending on your actual configuration.

Some expressions might refer to TCL functions or procedures (sometimes identified as `<functionof>`) that coreConsultant uses to make calculations. The exact formula used by these TCL functions is not provided in this chapter. However, when you configure the controller in coreConsultant, all TCL functions and parameters are evaluated completely; and the resulting values are displayed where appropriate in the coreConsultant GUI reports.

Exists Expressions

These expressions indicate the combination of configuration parameters required for a register, field, or block to exist in the memory map. The expression is only valid in the local context and does not indicate the conditions for existence of the parent. For example, the expression for a bit field in a register assumes that the register exists and does not include the conditions for existence of the register.

Offset

The term *Offset* is synonymous with *Address*.

Memory Access Attributes

The Memory Access attribute is defined as `<ReadBehavior>/<WriteBehavior>` which are defined in the following table.

Table 5-1 Possible Read and Write Behaviors

Read (or Write) Behavior	Description
RC	A read clears this register field.
RS	A read sets this register field.
RM	A read modifies the contents of this register field.
Wo	You can only write once to this register field.
W1C	A write of 1 clears this register field.
W1S	A write of 1 sets this register field.
W1T	A write of 1 toggles this register field.
W0C	A write of 0 clears this register field.
W0S	A write of 0 sets this register field.
W0T	A write of 0 toggles this register field.
WC	Any write clears this register field.
WS	Any write sets this register field.
WM	Any write toggles this register field.
no Read Behavior attribute	You cannot read this register. It is Write-Only.
no Write Behavior attribute	You cannot write to this register. It is Read-Only.

Table 5-2 Memory Access Examples

Memory Access	Description
R	Read-only register field.
W	Write-only register field.
R/W	Read/write register field.
R/W1C	You can read this register field. Writing 1 clears it.
RC/W1C	Reading this register field clears it. Writing 1 clears it.
R/Wo	You can read this register field. You can only write to it once.

Special Optional Attributes

Some register fields might use the following optional attributes.

Table 5-3 Optional Attributes

Attribute	Description
Volatile	As defined by the IP-XACT specification. If true, indicates in the case of a write followed by read, or in the case of two consecutive reads, there is no guarantee as to what is returned by the read on the second transaction or that this return value is consistent with the write or read of the first transaction. The element implies there is some additional mechanism by which this field can acquire new values other than by reads/writes/resets and other access methods known to IP-XACT. For example, when the controller updates the register field contents.
Testable	As defined by the IP-XACT specification. Possible values are unconstrained, untestable, readOnly, writeAsRead, restore. Untestable means that this field is untestable by a simple automated register test. For example, the read-write access of the register is controlled by a pin or another register. readOnly means that you should not write to this register; only read from it. This might apply for a register that modifies the contents of another register.
Reset Mask	As defined by the IP-XACT specification. Indicates that this register field has an unknown reset value. For example, the reset value is set by another register or an input pin; or the register is implemented using RAM.
* Varies	Indicates that the memory access (or reset) attribute (read, write behavior) is not fixed. For example, the read-write access of the register is controlled by a pin or another register. Or when the access depends on some configuration parameter; in this case the post-configuration report in coreConsultant gives the actual access value.

Register definitions for each component memory map.

Table 5-4 Registers for the DWC_mipi_dsi_host_MemMap Memory Map

Register	Offset	Description
DSI Registers DSI Exists: Always		
"VERSION" on page 190	0x0	Core Version.
"PWR_UP" on page 191	0x4	Power up.
"CLKMGR_CFG" on page 192	0x8	Factor for internal dividers.
"DPI_VCID" on page 193	0xc	DPI Virtual Channel ID.
"DPI_COLOR_CODING" on page 194	0x10	DPI color coding.
"DPI_CFG_POL" on page 196	0x14	DPI signals polarity.
"DPI_LP_CMD_TIM" on page 198	0x18	Low Power commands' timing.
"DBI_VCID" on page 199	0x1c	DBI Virtual Channel ID.

Register	Offset	Description
"DBI_CFG" on page 200	0x20	DBI pixels bit-width.
"DBI_PARTITIONING_EN" on page 202	0x24	DBI partition.
"DBI_CMDSIZE" on page 203	0x28	DBI command size.
"PCKHDL_CFG" on page 204	0x2c	EoTp, BTA, CRC and ECC Configuration.
"GEN_VCID" on page 206	0x30	Read responses Virtual Channel ID.
"MODE_CFG" on page 207	0x34	Operation mode.
"VID_MODE_CFG" on page 208	0x38	Video mode configuration.
"VID_PKT_SIZE" on page 211	0x3c	Video Packets size.
"VID_NUM_CHUNKS" on page 212	0x40	Number of Chunks.
"VID_NULL_SIZE" on page 213	0x44	Null Packets size.
"VID_HSA_TIME" on page 214	0x48	HSA time.
"VID_HBP_TIME" on page 215	0x4c	HBP time.
"VID_HLINE_TIME" on page 216	0x50	Overall video line time.
"VID_VSA_LINES" on page 217	0x54	VSA period.
"VID_VBP_LINES" on page 218	0x58	VBP period.
"VID_VFP_LINES" on page 219	0x5c	VFP period.
"VID_VACTIVE_LINES" on page 220	0x60	Video vertical resolution.
"EDPI_CMD_SIZE" on page 221	0x64	eDPI packets size.
"CMD_MODE_CFG" on page 222	0x68	Command Mode operation configuration.
"GEN_HDR" on page 226	0x6c	Generic Interface Packet Header.
"GEN_PLD_DATA" on page 227	0x70	Generic Interface Packets Payload.
"CMD_PKT_STATUS" on page 228	0x74	Generic Interface and DBI FIFO status.
"TO_CNT_CFG" on page 232	0x78	Timeout Trigger Configuration.
"HS_RD_TO_CNT" on page 234	0x7c	Peripheral timeout after HS read operations.
"LP_RD_TO_CNT" on page 235	0x80	Peripheral timeout after LP read operations.
"HS_WR_TO_CNT" on page 236	0x84	Peripheral timeout after HS write operations.
"LP_WR_TO_CNT" on page 237	0x88	Peripheral timeout after LP write operations.
"BTA_TO_CNT" on page 238	0x8c	Peripheral timeout after BTA completion.
"SDF_3D" on page 239	0x90	3D information for VSS packets.

Register	Offset	Description
"LPCLK_CTRL" on page 241	0x94	Non-continuous Clock configuration.
"PHY_TMR_LPCLK_CFG" on page 242	0x98	Time configuration for (clock lane) transitions between HS and LP.
"PHY_TMR_CFG" on page 243	0x9c	Time configuration for (data lanes) transitions between HS and LP.
"PHY_RSTZ" on page 244	0xa0	D-PHY's PLL and Resets.
"PHY_IF_CFG" on page 245	0xa4	Active lanes and Stop State minimum time in Stop State.
"PHY_ULPS_CTRL" on page 246	0xa8	Transitions from and to ULPS, using D-PHY.
"PHY_TX_TRIGGER" on page 247	0xac	Pins related to D-PHY triggers.
"PHY_STATUS" on page 248	0xb0	D-PHY Status.
"PHY_TST_CTRL0" on page 251	0xb4	D-PHY control and clear pins.
"PHY_TST_CTRL1" on page 252	0xb8	D-PHY data and enable pins.
"INT_ST0" on page 253	0xbc	Interrupts status 0.
"INT_ST1" on page 256	0xc0	Interrupts Status 1.
"INT_MSK0" on page 259	0xc4	INT_ST0 mask.
"INT_MSK1" on page 262	0xc8	INT_ST1 mask.
"PHY_CAL" on page 265	0xcc	D-PHY skew calibration.
"INT_FORCE0" on page 266	0xd8	Force INT_ST0
"INT_FORCE1" on page 269	0xe0	Force INT_ST1.
"AUTO_ULPS_MODE" on page 272	0xe0	Automatic ULPS control.
"AUTO_ULPS_ENTRY_DELAY" on page 273	0xe4	ULPS transition delay.
"AUTO_ULPS_WAKEUP_TIME" on page 274	0xe8	D-PHY wakeup time.
"DSC_PARAMETER" on page 275	0xf0	Display Stream Compression.
"PHY_TMR_RD_CFG" on page 277	0xf4	PHY timings.
"AUTO_ULPS_MIN_TIME" on page 278	0xf8	PHY Timings - Transition between ulpsactivenot and ulpsexitreq.
"PHY_MODE" on page 279	0xfc	PHY interface.
"VID_SHADOW_CTRL" on page 280	0x100	DPI Shadow Feature.
"DPI_VCID_ACT" on page 281	0x10c	Actual DPI Virtual Channel ID.
"DPI_COLOR_CODING_ACT" on page 282	0x110	Actual DPI Color Coding

Register	Offset	Description
"DPI_LP_CMD_TIM_ACT" on page 284	0x118	Actual DPI Low Power Commands' Timing.
"EDPI_TE_HW_CFG" on page 285	0x11c	TE for Hardware operations.
"VID_MODE_CFG_ACT" on page 287	0x138	VID_MODE_CFG.
"VID_PKT_SIZE_ACT" on page 289	0x13c	Actual VID_PKT_SIZE.
"VID_NUM_CHUNKS_ACT" on page 290	0x140	Actual VID_NUM_CHUNKS.
"VID_NULL_SIZE_ACT" on page 291	0x144	Actual VID_NULL_SIZE.
"VID_HSA_TIME_ACT" on page 292	0x148	Actual VID_HSA_TIME.
"VID_HBP_TIME_ACT" on page 293	0x14c	Actual VID_HBP_TIME.
"VID_HLINE_TIME_ACT" on page 294	0x150	Actual VID_HLINE_TIME.
"VID_VSA_LINES_ACT" on page 295	0x154	Actual VID_VSA_LINES.
"VID_VBP_LINES_ACT" on page 296	0x158	VID_VBP_LINES.
"VID_VFP_LINES_ACT" on page 297	0x15c	Actual VID_VFP_LINES.
"VID_VACTIVE_LINES_ACT" on page 298	0x160	Actual VID_VACTIVE_LINES.
"VID_PKT_STATUS" on page 299	0x168	eDPI and DPI FIFOs status.
"SDF_3D_ACT" on page 302	0x190	SDF 3D.
"DSC_ENC_COREID" on page 304	0x200	COREID.
"DSC_ENC_VERSION" on page 305	0x204	DSC Version.
"DSC_ENC_FLATNESS_DET_THRES" on page 306	0x208	DSC encoder flatness.
"DSC_ENC_DELAY" on page 307	0x20c	DPI events delay.
"DSC_ENC_COMPRESSED_LINE_SIZE" on page 308	0x210	Compressed line size.
"DSC_ENC_LINES_IN_EXCESS" on page 309	0x214	Number of lines in excess.
"DSC_ENC_RBUF_ADDR_LAST_LINE_ADJ" on page 310	0x218	DSC adjustment.
"DSC_MODE" on page 311	0x21c	DSC encoder.
"DSC_ENC_INT_ST" on page 312	0x220	DSC encoder interrupt status.
"DSC_ENC_INT_MSK" on page 315	0x224	INT_ST_DSC mask.
"DSC_ENC_INT_FORCE" on page 318	0x228	INT_ST_DSC interrupt force.
"DSC_FIFO_STATUS_SELECT" on page 321	0x22c	DSC FIFO selection.

Register	Offset	Description
"DSC_FIFO_STATUS" on page 322	0x230	DSC encoder FIFOs status.
"DSC_FIFO_STATUS2" on page 324	0x234	Second DSC encoder FIFOs status.
"DSC_FIFO_WORD_COUNT" on page 326	0x238	DSC encoder FIFOs word count.
"DSC_FIFO_WORD_COUNT2" on page 327	0x23c	Second DSC encoder FIFOs word count.
"DSC_ENC_PPS_0_3" on page 328	0x260	Picture Parameter Set[0,3].
"DSC_ENC_PPS_4_7" on page 330	0x264	Picture Parameter Set[4,7].
"DSC_ENC_PPS_8_11" on page 332	0x268	Picture Parameter Set[8,11].
"DSC_ENC_PPS_12_15" on page 333	0x26c	Picture Parameter Set[12,15].
"DSC_ENC_PPS_16_19" on page 334	0x270	Picture Parameter Set[16,19].
"DSC_ENC_PPS_20_23" on page 335	0x274	Picture Parameter Set[20,23].
"DSC_ENC_PPS_24_27" on page 336	0x278	Picture Parameter Set[24,27].
"DSC_ENC_PPS_28_31" on page 337	0x27c	Picture Parameter Set[28,31].
"DSC_ENC_PPS_32_35" on page 339	0x280	Picture Parameter Set[32,35].
"DSC_ENC_PPS_36_39" on page 340	0x284	Picture Parameter Set[36,39].
"DSC_ENC_PPS_40_43" on page 341	0x288	Picture Parameter Set[40,43].
"DSC_ENC_PPS_44_47" on page 343	0x28c	Picture Parameter Set[44,47].
"DSC_ENC_PPS_48_51" on page 345	0x290	Picture Parameter Set[48,51].
"DSC_ENC_PPS_52_55" on page 347	0x294	Picture Parameter Set[52,55].
"DSC_ENC_PPS_56_59" on page 349	0x298	Picture Parameter Set[56,59].
"DSC_ENC_PPS_60_63" on page 351	0x29c	Picture Parameter Set[60,63].
"DSC_ENC_PPS_64_67" on page 353	0x2a0	Picture Parameter Set[64,67].
"DSC_ENC_PPS_68_71" on page 355	0x2a4	Picture Parameter Set[68,71].
"DSC_ENC_PPS_72_75" on page 357	0x2a8	Picture Parameter Set[72,75].
"DSC_ENC_PPS_76_79" on page 359	0x2ac	Picture Parameter Set[76,79].
"DSC_ENC_PPS_80_83" on page 361	0x2b0	Picture Parameter Set[80,83].
"DSC_ENC_PPS_84_87" on page 363	0x2b4	Picture Parameter Set[84,87].
"INT_ST0_AP" on page 365	0x300	Internal Diagnosis Group 0.
"INT_MSK0_AP" on page 368	0x304	Mask for INT_ST0_AP.
"INT_FORCE0_AP" on page 371	0x308	Force to INT_ST0_AP.

Register	Offset	Description
"INT_ST1_AP" on page 374	0x310	Internal Diagnosis Group 1.
"INT_MSK1_AP" on page 377	0x314	Mask for INT_ST1_AP.
"INT_FORCE1_AP" on page 379	0x318	Force to INT_ST1_AP
"INT_ST2_AP" on page 381	0x320	Internal Diagnosis Group 2.
"INT_MSK2_AP" on page 384	0x324	Mask for INT_ST2_AP.
"INT_FORCE2_AP" on page 387	0x328	Force to INT_ST2_AP.
"TO_HSTXRDY_CFG_AP" on page 390	0x340	HS TX RDY timeout.
"TO_LPTXRDY_CFG_AP" on page 391	0x344	LP TX RDY timeout
"TO_LPTXTRIG_CFG_AP" on page 392	0x348	LP TX TRIG timeout.
"TO_LPTXULPS_CFG_AP" on page 393	0x34c	LP TX ULPS timeout.
"TO_HSTX_CFG_AP" on page 394	0x350	HS TX timeout.
"TO_LPRX_CFG_AP" on page 395	0x354	LP RX timeout.
"TO_BTA_CFG_AP" on page 396	0x358	Counter for BTA timeout.
"TO_CLK_DIV_AP" on page 397	0x35c	Factor for internal clock dividers.
"ERR_INJ_CTRL_AP" on page 398	0x380	FMEDA memory error injection.
"ERR_INJ_CHK_MSK_AP" on page 399	0x384	Mask for FMEDA ECC check bits.
"ERR_INJ_DATA_MSK_AP" on page 400	0x388	Mask for FMEDA 32-bit data error injection.
"ERR_INJ_ST_AP" on page 401	0x38c	FMEDA mode error injection status.

5.1 DSI Registers

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.1 VERSION

- **Name:** Core Version.
- **Description:** Contains the version of the DSI host controller.
- **Size:** 32 bits
- **Offset:** 0x0
- **Exists:** Always

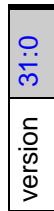


Table 5-5 Fields for Register: VERSION

Bits	Name	Memory Access	Description
31:0	version	R	This field indicates the version of the DWC_mipi_dsi_host. Value After Reset: DSI_HOST_VERSION_ID Exists: Always

5.1.2 PWR_UP

- **Name:** Power up.
- **Description:** Controls the power up of the controller.
- **Size:** 32 bits
- **Offset:** 0x4
- **Exists:** Always

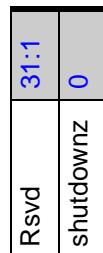


Table 5-6 Fields for Register: PWR_UP

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	shutdownz	R/W	<p>This bit configures the controller either to power up or to reset.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (RESET): Reset the controller ■ 0x1 (POWERUP): Power up the controller <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.3 CLKMGR_CFG

- **Name:** Factor for internal dividers.
- **Description:** Configures the factor for internal dividers to divide lanebyteclk for timeout purposes.
- **Size:** 32 bits
- **Offset:** 0x8
- **Exists:** Always

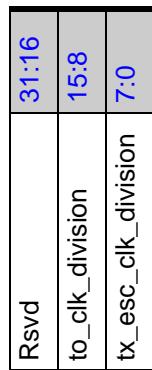


Table 5-7 Fields for Register: CLKMGR_CFG

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:8	to_clk_division	R/W	This field indicates the division factor for the Time Out clock used as the timing unit in the configuration of high-speed to low-power and low-power to high-speed transition error. Value After Reset: 0x0 Exists: Always
7:0	tx_esc_clk_division	R/W	This field indicates the division factor for the TX Escape clock source (lanebyteclk). The values 0 and 1 stop the TX_ESC clock generation. Value After Reset: 0x0 Exists: Always

5.1.4 DPI_VCID

- **Name:** DPI Virtual Channel ID.
- **Description:** Configures the Virtual Channel ID for DPI traffic.
- **Size:** 32 bits
- **Offset:** 0xc
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

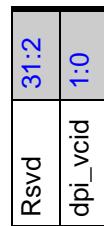


Table 5-8 Fields for Register: DPI_VCID

Bits	Name	Memory Access	Description
31:2			Reserved Field: Yes
1:0	dpi_vcid	R/W	This field configures the DPI virtual channel id that is indexed to the Video mode packets. Value After Reset: 0x0 Exists: Always

5.1.5 DPI_COLOR_CODING

- **Name:** DPI color coding.
- **Description:** Configures DPI color coding.
- **Size:** 32 bits
- **Offset:** 0x10
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

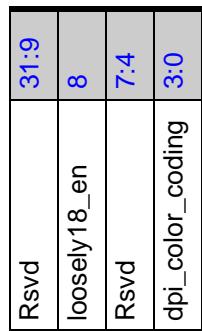


Table 5-9 Fields for Register: DPI_COLOR_CODING

Bits	Name	Memory Access	Description
31:9			Reserved Field: Yes
8	loosely18_en	R/W	When set to 1, this bit activates loosely packed variant to 18-bit configurations. Value After Reset: 0x0 Exists: Always
7:4			Reserved Field: Yes

Bits	Name	Memory Access	Description
3:0	dpi_color_coding	R/W	<p>This field configures the DPI color for Video Mode/eDPI Command Mode coding as follows:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (CC00): 16-bit configuration 1 ■ 0x1 (CC01): 16-bit configuration 2 ■ 0x2 (CC02): 16-bit configuration 3 ■ 0x3 (CC03): 18-bit configuration 1 ■ 0x4 (CC04): 18-bit configuration 2 ■ 0x5 (CC05): 24-bit ■ 0x6 (CC06): 20-bit YCbCr 4:2:2 loosely packed / Reserved for eDPI Command Mode ■ 0x7 (CC07): 24-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode ■ 0x8 (CC08): 16-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode ■ 0x9 (CC09): 30-bit - DSC_ENC 10bit / Reserved for eDPI Command Mode ■ 0xa (CC10): 36-bit / Reserved for eDPI Command Mode ■ 0xb (CC11): 12-bit YCbCr 4:2:0 / Reserved for eDPI Command Mode ■ 0xc (CC12): DSC24 compressed Data ■ 0xd (CC13): Reserved ■ 0xe (CC14): Reserved ■ 0xf (CC15): Reserved <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.6 DPI_CFG_POL

- **Name:** DPI signals polarity.
- **Description:** Configures the polarity of DPI signals.
- **Size:** 32 bits
- **Offset:** 0x14
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

Rsvd	31:5	4	3	2	1	0
		colorm_active_low	shutd_active_low	hsync_active_low	vsync_active_low	dataen_active_low

Table 5-10 Fields for Register: DPI_CFG_POL

Bits	Name	Memory Access	Description
31:5			Reserved Field: Yes
4	colorm_active_low	R/W	When set to 1, this bit configures the color mode pin (dpicolorm) as active low. Value After Reset: 0x0 Exists: Always
3	shutd_active_low	R/W	When set to 1, this bit configures the shutdown pin (dpishutdn) as active low. Value After Reset: 0x0 Exists: Always
2	hsync_active_low	R/W	When set to 1, this bit configures the horizontal synchronism pin (dipihsync) as active low. Value After Reset: 0x0 Exists: Always
1	vsync_active_low	R/W	When set to 1, this bit configures the vertical synchronism pin (dipivsync) as active low. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
0	dataen_active_low	R/W	<p>When set to 1, this bit configures the data enable pin (dpidataen) as active low.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.7 DPI_LP_CMD_TIM

- **Name:** Low Power commands' timing.
- **Description:** Configures the timing for low-power commands sent while in video mode.
- **Size:** 32 bits
- **Offset:** 0x18
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

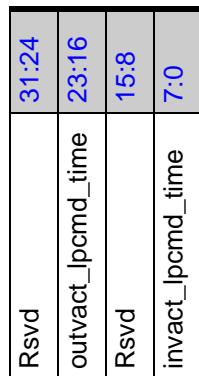


Table 5-11 Fields for Register: DPI_LP_CMD_TIM

Bits	Name	Memory Access	Description
31:24			Reserved Field: Yes
23:16	outvact_lpcmd_time	R/W	This field is used for the transmission of commands in low-power mode. It defines the size, in bytes, of the largest packet that can fit in a line during the VSA, VBP, and VFP regions. Value After Reset: 0x0 Exists: Always
15:8			Reserved Field: Yes
7:0	invact_lpcmd_time	R/W	This field is used for the transmission of commands in low-power mode. It defines the size, in bytes, of the largest packet that can fit in a line during the VACT region. Value After Reset: 0x0 Exists: Always

5.1.8 DBI_VCID

- **Name:** DBI Virtual Channel ID.
- **Description:** Configures Virtual Channel ID for DBI traffic.
- **Size:** 32 bits
- **Offset:** 0x1c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 1) || (DSI_HOST_DATAINTERFACE == 3)) == 1

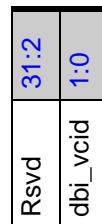


Table 5-12 Fields for Register: DBI_VCID

Bits	Name	Memory Access	Description
31:2			Reserved Field: Yes
1:0	dbi_vcidx	R/W	This field configures the virtual channel id that is indexed to the DCS packets from DBI. Value After Reset: 0x0 Exists: Always

5.1.9 DBI_CFG

- **Name:** DBI pixels bit-width.
- **Description:** Configures the bit width of pixels for DBI.
- **Size:** 32 bits
- **Offset:** 0x20
- **Exists:** ((DSI_HOST_DATAINTERFACE == 1) || (DSI_HOST_DATAINTERFACE == 3)) == 1

Rsvd	31:18
lut_size_conf	17:16
Rsvd	15:12
out_dbi_conf	11:8
Rsvd	7:4
in_dbi_conf	3:0

Table 5-13 Fields for Register: DBI_CFG

Bits	Name	Memory Access	Description
31:18			Reserved Field: Yes
17:16	lut_size_conf	R/W	<p>This field configures the size used to transport the Write Lut commands as follows:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (LUTSIZE0): 16-bit color display ■ 0x1 (LUTSIZE1): 18-bit color display ■ 0x2 (LUTSIZE2): 24-bit color display ■ 0x3 (LUTSIZE3): 32-bit color display <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:12			Reserved Field: Yes

Bits	Name	Memory Access	Description
11:8	out_dbi_conf	R/W	<p>This field configures the DBI output pixel data as follows:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (PIXELDATA0): 8-bit 8 bpp ■ 0x1 (PIXELDATA1): 8-bit 12 bpp ■ 0x2 (PIXELDATA2): 8-bit 16 bpp ■ 0x3 (PIXELDATA3): 8-bit 18 bpp ■ 0x4 (PIXELDATA4): 8-bit 24 bpp ■ 0x5 (PIXELDATA5): 9-bit 18 bpp ■ 0x6 (PIXELDATA6): 16-bit 8 bpp ■ 0x7 (PIXELDATA7): 16-bit 12 bpp ■ 0x8 (PIXELDATA8): 16-bit 16 bpp ■ 0x9 (PIXELDATA9): 16-bit 18 bpp, option 1 ■ 0xa (PIXELDATA10): 16-bit 18 bpp, option 2 ■ 0xb (PIXELDATA11): 16-bit 24 bpp, option 1 ■ 0xc (PIXELDATA12): 16-bit 24 bpp, option 2 <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7:4			Reserved Field: Yes
3:0	in_dbi_conf	R/W	<p>This field configures the DBI input pixel data as follows:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (PIXELDATA0): 8-bit 8 bpp ■ 0x1 (PIXELDATA1): 8-bit 12 bpp ■ 0x2 (PIXELDATA2): 8-bit 16 bpp ■ 0x3 (PIXELDATA3): 8-bit 18 bpp ■ 0x4 (PIXELDATA4): 8-bit 24 bpp ■ 0x5 (PIXELDATA5): 9-bit 18 bpp ■ 0x6 (PIXELDATA6): 16-bit 8 bpp ■ 0x7 (PIXELDATA7): 16-bit 12 bpp ■ 0x8 (PIXELDATA8): 16-bit 16 bpp ■ 0x9 (PIXELDATA9): 16-bit 18 bpp, option 1 ■ 0xa (PIXELDATA10): 16-bit 18 bpp, option 2 ■ 0xb (PIXELDATA11): 16-bit 24 bpp, option 1 ■ 0xc (PIXELDATA12): 16-bit 24 bpp, option 2 <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.10 DBI_PARTITIONING_EN

- **Name:** DBI partition.
- **Description:** Configures whether DWC_mipi_dsi_host is to partition DBI traffic automatically.
- **Size:** 32 bits
- **Offset:** 0x24
- **Exists:** ((DSI_HOST_DATAINTERFACE == 1) || (DSI_HOST_DATAINTERFACE == 3)) == 1



Table 5-14 Fields for Register: DBI_PARTITIONING_EN

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	partitioning_en	R/W	<p>When set to 1, this bit enables the use of write_memory_continue input commands (system needs to ensure correct partitioning of Long Write commands). When not set, partitioning is automatically performed in the DWC_mipi_dsi_host.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.11 DBI_CMDSIZE

- **Name:** DBI command size.
- **Description:** Configures the command size and the size for automatic partitioning of DBI packets.
- **Size:** 32 bits
- **Offset:** 0x28
- **Exists:** ((DSI_HOST_DATAINTERFACE == 1) || (DSI_HOST_DATAINTERFACE == 3)) == 1



Table 5-15 Fields for Register: DBI_CMDSIZE

Bits	Name	Memory Access	Description
31:16	allowed_cmd_size	R/W	<p>This field configures the maximum allowed size for a DCS write memory command. This field is used to partition a write memory command into one write_memory_start and a variable number of write_memory_continue commands. It is only used if the partitioning_en bit of the DBI_CFG register is disabled. The size of the DSI packet payload is the actual payload size minus 1, because the DCS command is in the DSI packet payload.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:0	wr_cmd_size	R/W	<p>This field configures the size of the DCS write memory commands. The size of DSI packet payload is the actual payload size minus 1, because the DCS command is in the DSI packet payload.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.12 PCKHDL_CFG

- **Name:** EoTp, BTA, CRC and ECC Configuration.
- **Description:** Configures how EoTp, BTA, CRC and ECC are to be used, to meet peripherals characteristics
- **Size:** 32 bits
- **Offset:** 0x2c
- **Exists:** Always

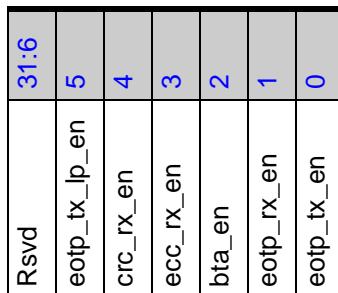


Table 5-16 Fields for Register: PCKHDL_CFG

Bits	Name	Memory Access	Description
31:6			Reserved Field: Yes
5	eotp_tx_lp_en	R/W	When set to 1, this bit enables the EoTp transmission in low-power. Value After Reset: 0x0 Exists: Always
4	crc_rx_en	R/W	When set to 1, this bit enables the CRC reception and error reporting. Value After Reset: 0x0 Exists: Always
3	ecc_rx_en	R/W	When set to 1, this bit enables the ECC reception, error correction, and reporting. Value After Reset: 0x0 Exists: Always
2	bta_en	R/W	When set to 1, this bit enables the Bus Turn-Around (BTA) request. Value After Reset: 0x0 Exists: Always
1	eotp_rx_en	R/W	When set to 1, this bit enables the EoTp reception. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
0	eotp_tx_en	R/W	<p>When set to 1, this bit enables the EoTp transmission in high-speed.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.13 GEN_VCID

- **Name:** Read responses Virtual Channel ID.
- **Description:** Configures the Virtual Channel ID of READ responses to store and return to Generic interface.
- **Size:** 32 bits
- **Offset:** 0x30
- **Exists:** Always

Rsvd	31:18
gen_vcid_tx_auto	17:16
Rsvd	15:10
gen_vcid_tear_auto	9:8
Rsvd	7:2
gen_vcid_rx	1:0

Table 5-17 Fields for Register: GEN_VCID

Bits	Name	Memory Access	Description
31:18			Reserved Field: Yes
17:16	gen_vcid_tx_auto	R/W	This field indicates the Generic interface virtual channel identification where generic packet is automatically generated & transmitted. Value After Reset: 0x0 Exists: Always
15:10			Reserved Field: Yes
9:8	gen_vcid_tear_auto	R/W	This field indicates the virtual channel identification for tear effect by hardware Value After Reset: 0x0 Exists: Always
7:2			Reserved Field: Yes
1:0	gen_vcid_rx	R/W	This field indicates the Generic interface read-back virtual channel identification. Value After Reset: 0x0 Exists: Always

5.1.14 MODE_CFG

- **Name:** Operation mode.
- **Description:** Configures the mode of operation between Video or Command Mode. (Commands can still be sent while in video mode.)
- **Size:** 32 bits
- **Offset:** 0x34
- **Exists:** Always

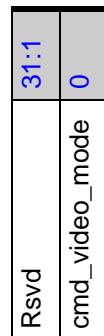


Table 5-18 Fields for Register: MODE_CFG

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	cmd_video_mode	R/W	<p>This bit configures the operation mode:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (VIDMODE): video mode ■ 0x1 (CMDMODE): command mode <p>Value After Reset: 0x1</p> <p>Exists: Always</p>

5.1.15 VID_MODE_CFG

- **Name:** Video mode configuration.
- **Description:** Configures several aspects of Video mode operation, the transmission mode, switching to low-power in the middle of a frame, enabling acknowledge and whether to send commands in low-power.
- **Size:** 32 bits
- **Offset:** 0x38
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

	31:25	Rsvd	vpg_orientation	24	Rsvd	vpg_mode	20	Rsvd	19:17	vpg_en	16	lp_cmd_en	15	frame_bta_ack_en	14	lp_hfp_en	13	lp_hbp_en	12	lp_vact_en	11	lp_vfp_en	10	lp_vbp_en	9	lp_vsa_en	8	Rsvd	7:2	vid_mode_type	1:0
--	-------	------	-----------------	----	------	----------	----	------	-------	--------	----	-----------	----	------------------	----	-----------	----	-----------	----	------------	----	-----------	----	-----------	---	-----------	---	------	-----	---------------	-----

Table 5-19 Fields for Register: VID_MODE_CFG

Bits	Name	Memory Access	Description
31:25			Reserved Field: Yes
24	vpg_orientation	R/W	<p>This field indicates the color bar orientation as follows: Values:</p> <ul style="list-style-type: none"> ■ 0x0 (VPGORENT0): Vertical mode ■ 0x1 (VPGORENT1): Horizontal mode <p>Value After Reset: 0x0 Exists: Always</p>
23:21			Reserved Field: Yes
20	vpg_mode	R/W	<p>This field is to select the pattern: Values:</p> <ul style="list-style-type: none"> ■ 0x0 (COLORBAR): horizontal or vertical ■ 0x1 (BERPATTERN): vertical only <p>Value After Reset: 0x0 Exists: Always</p>
19:17			Reserved Field: Yes

Bits	Name	Memory Access	Description
16	vpg_en	R/W	<p>When set to 1, this bit enables the video mode pattern generator.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15	lp_cmd_en	R/W	<p>When set to 1, this bit enables the command transmission only in low-power mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
14	frame_bta_ack_en	R/W	<p>When set to 1, this bit enables the request for an acknowledgement response at the end of a frame.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	lp_hfp_en	R/W	<p>When set to 1, this bit enables the return to low-power inside the HFP period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
12	lp_hbp_en	R/W	<p>When set to 1, this bit enables the return to low-power inside the HBP period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	lp_vact_en	R/W	<p>When set to 1, this bit enables the return to low-power inside the VACT period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	lp_vfp_en	R/W	<p>When set to 1, this bit enables the return to low-power inside the VFP period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	lp_vbp_en	R/W	<p>When set to 1, this bit enables the return to low-power inside the VBP period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	lp_vsa_en	R/W	<p>When set to 1, this bit enables the return to low-power inside the VSA period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7:2			Reserved Field: Yes

Bits	Name	Memory Access	Description
1:0	vid_mode_type	R/W	<p>This field indicates the video mode transmission type as follows:</p> <p>Values:</p> <ul style="list-style-type: none">■ 0x0 (VIDMODE0): Non-burst with sync pulses■ 0x1 (VIDMODE1): Non-burst with sync events■ 0x2 (VIDMODE2): Burst mode■ 0x3 (VIDMODE3): Burst mode <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.16 VID_PKT_SIZE

- **Name:** Video Packets size.
- **Description:** Configures the video packet size.
- **Size:** 32 bits
- **Offset:** 0x3c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

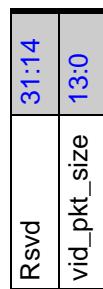


Table 5-20 Fields for Register: VID_PKT_SIZE

Bits	Name	Memory Access	Description
31:14			Reserved Field: Yes
13:0	vid_pkt_size	R/W	<p>This field configures the number of pixels in a single video packet.</p> <ul style="list-style-type: none"> ■ For 18-bit not loosely packed data types, this number must be a multiple of 4. ■ For YCbCr data types, it must be a multiple of 2, as described in the DSI specification. ■ For DSC24, this field configures the number of Bytes. Note that when using slices (i.e chunks) the minimum size packet size must be 3. <p>Value After Reset: 0x0 Exists: Always</p>

5.1.17 VID_NUM_CHUNKS

- **Name:** Number of Chunks.
- **Description:** Configures the number of chunks to use. The data in each chunk has the size provided by VID_PKT_SIZE.
- **Size:** 32 bits
- **Offset:** 0x40
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1



Table 5-21 Fields for Register: VID_NUM_CHUNKS

Bits	Name	Memory Access	Description
31:13			Reserved Field: Yes
12:0	vid_num_chunks	R/W	<p>This register configures the number of chunks to be transmitted during a Line period (a chunk is pair made of a video packet and a null packet). If set to 0 or 1, video line is still transmitted in a single packet. If set to 1 that packet is part of a chunk, meaning that a null packet follows it (if vid_null_size > 0). Otherwise, multiple chunks are used to transmit each video line.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.18 VID_NULL_SIZE

- **Name:** Null Packets size.
- **Description:** Configures the size of null packets.
- **Size:** 32 bits
- **Offset:** 0x44
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

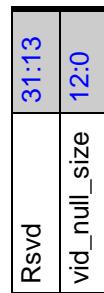


Table 5-22 Fields for Register: VID_NULL_SIZE

Bits	Name	Memory Access	Description
31:13			Reserved Field: Yes
12:0	vid_null_size	R/W	This register configures the number of bytes inside a null packet. Setting to 0 disables null packets. Value After Reset: 0x0 Exists: Always

5.1.19 VID_HSA_TIME

- **Name:** HSA time.
- **Description:** Configures the video HSA time.
- **Size:** 32 bits
- **Offset:** 0x48
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

Rsvd	31:12
vid_hsa_time	11:0

Table 5-23 Fields for Register: VID_HSA_TIME

Bits	Name	Memory Access	Description
31:12			Reserved Field: Yes
11:0	vid_hsa_time	R/W	This field configures the Horizontal Synchronization Active period in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.20 VID_HBP_TIME

- **Name:** HBP time.
- **Description:** Configures the video HBP time.
- **Size:** 32 bits
- **Offset:** 0x4c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

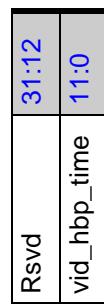


Table 5-24 Fields for Register: VID_HBP_TIME

Bits	Name	Memory Access	Description
31:12			Reserved Field: Yes
11:0	vid_hbp_time	R/W	This field configures the Horizontal Back Porch period in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.21 VID_HLINE_TIME

- **Name:** Overall video line time.
- **Description:** Configures the overall time for each video line.
- **Size:** 32 bits
- **Offset:** 0x50
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

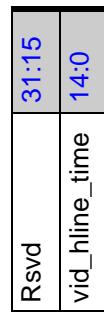


Table 5-25 Fields for Register: VID_HLINE_TIME

Bits	Name	Memory Access	Description
31:15			Reserved Field: Yes
14:0	vid_hline_time	R/W	This field configures the size of the total line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.22 VID_VSA_LINES

- **Name:** VSA period.
- **Description:** Configures the VSA period.
- **Size:** 32 bits
- **Offset:** 0x54
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

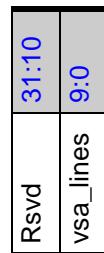


Table 5-26 Fields for Register: VID_VSA_LINES

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9:0	vsa_lines	R/W	This field configures the Vertical Synchronization Active period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.23 VID_VBP_LINES

- **Name:** VBP period.
- **Description:** Configures the VBP period.
- **Size:** 32 bits
- **Offset:** 0x58
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

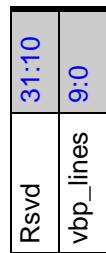


Table 5-27 Fields for Register: VID_VBP_LINES

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9:0	vbp_lines	R/W	This field configures the Vertical Back Porch period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.24 VID_VFP_LINES

- **Name:** VFP period.
- **Description:** Configures the VFP period.
- **Size:** 32 bits
- **Offset:** 0x5c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

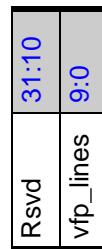


Table 5-28 Fields for Register: VID_VFP_LINES

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9:0	vfp_lines	R/W	This field configures the Vertical Front Porch period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.25 VID_VACTIVE_LINES

- **Name:** Video vertical resolution.
- **Description:** Configures the vertical resolution of video.
- **Size:** 32 bits
- **Offset:** 0x60
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

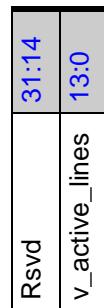


Table 5-29 Fields for Register: VID_VACTIVE_LINES

Bits	Name	Memory Access	Description
31:14			Reserved Field: Yes
13:0	v_active_lines	R/W	This field configures the Vertical Active period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.26 EDPI_CMD_SIZE

- **Name:** eDPI packets size.
- **Description:** Configures the size of eDPI packets.
- **Size:** 32 bits
- **Offset:** 0x64
- **Exists:** (DSI_HOST_DATAINTERFACE == 4)==1

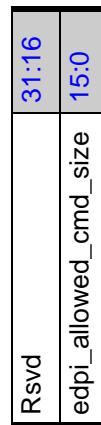


Table 5-30 Fields for Register: EDPI_CMD_SIZE

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	edpi_allowed_cmd_size	R/W	This field configures the maximum allowed size for an eDPI write memory command, measured in pixels. Automatic partitioning of data obtained from eDPI is permanently enabled. Value After Reset: 0x0 Exists: Always

5.1.27 CMD MODE CFG

- **Name:** Command Mode operation configuration.
 - **Description:** Configures several aspect of command mode operation, tearing effect, acknowledge for each packet and the speed mode to transmit each Data Type related to commands.
 - **Size:** 32 bits
 - **Offset:** 0x68
 - **Exists:** Always

Rsvd	31:25
max_rd_pkt_size	24
Rsvd	23:20
dcs_hw_tx	19
dcs_sr_0p_tx	18
dcs_sw_1p_tx	17
dcs_sw_0p_tx	16
Rsvd	15
gen_hw_tx	14
gen_sr_2p_tx	13
gen_sr_1p_tx	12
gen_sr_0p_tx	11
gen_sw_2p_tx	10
gen_sw_1p_tx	9
gen_sw_0p_tx	8
Rsvd	7:2
ack_rqst_en	1
tear_fx_en	0

Table 5-31 Fields for Register: CMD_MODE_CFG

Bits	Name	Memory Access	Description
31:25			Reserved Field: Yes
24	max_rd_pkt_size	R/W	<p>This bit configures the maximum read packet size command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
23:20			Reserved Field: Yes
19	dcs_lw_tx	R/W	<p>This bit configures the DCS long write packet command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
18	dcs_sr_0p_tx	R/W	<p>This bit configures the DCS short read packet with zero parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
17	dcs_sw_1p_tx	R/W	<p>This bit configures the DCS short write packet with one parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
16	dcs_sw_0p_tx	R/W	<p>This bit configures the DCS short write packet with zero parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15			Reserved Field: Yes
14	gen_lw_tx	R/W	<p>This bit configures the Generic long write packet command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	gen_sr_2p_tx	R/W	<p>This bit configures the Generic short read packet with two parameters command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
12	gen_sr_1p_tx	R/W	<p>This bit configures the Generic short read packet with one parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	gen_sr_0p_tx	R/W	<p>This bit configures the Generic short read packet with zero parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	gen_sw_2p_tx	R/W	<p>This bit configures the Generic short write packet with two parameters command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	gen_sw_1p_tx	R/W	<p>This bit configures the Generic short write packet with one parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	gen_sw_0p_tx	R/W	<p>This bit configures the Generic short write packet with zero parameter command transmission type:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (HIGHSPEED): Transition type is high-speed ■ 0x1 (LOWPOWER): Transition type is low-power <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7:2			Reserved Field: Yes
1	ack_rqst_en	R/W	<p>When set to 1, this bit enables the acknowledge request after each packet transmission.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
0	tear_fx_en	R/W	<p>When set to 1, this bit enables the tearing effect acknowledge request.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.28 GEN_HDR

- **Name:** Generic Interface Packet Header.
- **Description:** Sets the header for new packets sent using the Generic interface.
- **Size:** 32 bits
- **Offset:** 0x6c
- **Exists:** Always

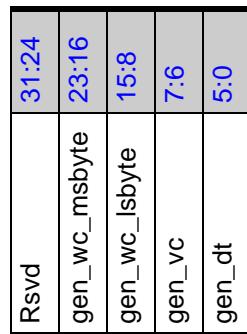


Table 5-32 Fields for Register: GEN_HDR

Bits	Name	Memory Access	Description
31:24			Reserved Field: Yes
23:16	gen_wc_msbyte	R/W	<p>This field configures the most significant byte of the header packet's word count for long packets or data 1 for short packets.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:8	gen_wc_lsbyte	R/W	<p>This field configures the least significant byte of the header packet's Word count for long packets or data 0 for short packets.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7:6	gen_vc	R/W	<p>This field configures the Virtual Channel ID of the header packet.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5:0	gen_dt	R/W	<p>This field configures the packet Data Type of the header packet.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.29 GEN_PLD_DATA

- **Name:** Generic Interface Packets Payload.
- **Description:** Sets the payload for packets sent using the Generic interface and, when read returns the contents of READ responses from the peripheral.
- **Size:** 32 bits
- **Offset:** 0x70
- **Exists:** Always

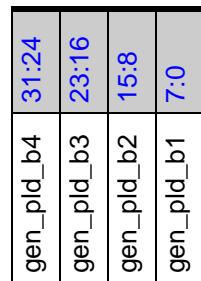


Table 5-33 Fields for Register: GEN_PLD_DATA

Bits	Name	Memory Access	Description
31:24	gen_pld_b4	R/W	This field indicates byte 4 of the packet payload. Value After Reset: 0x0 Exists: Always
23:16	gen_pld_b3	R/W	This field indicates byte 3 of the packet payload. Value After Reset: 0x0 Exists: Always
15:8	gen_pld_b2	R/W	This field indicates byte 2 of the packet payload. Value After Reset: 0x0 Exists: Always
7:0	gen_pld_b1	R/W	This field indicates byte 1 of the packet payload. Value After Reset: 0x0 Exists: Always

5.1.30 CMD PKT STATUS

- **Name:** Generic Interface and DBI FIFO status.
 - **Description:** Contains information about the status of FIFOs related to DBI and Generic interface.
 - **Size:** 32 bits
 - **Offset:** 0x74
 - **Exists:** Always

Rsvd	31:28
dbi_buff_pld_full	27
dbi_buff_pld_empty	26
dbi_buff_cmd_full	25
dbi_buff_cmd_empty	24
Rsvd	23:20
gen_buff_pld_full	19
gen_buff_pld_empty	18
gen_buff_cmd_full	17
gen_buff_cmd_empty	16
Rsvd	15
dbi_rd_cmd_busy	14
dbi_pld_r_full	13
dbi_pld_r_empty	12
dbi_pld_w_full	11
dbi_pld_w_empty	10
dbi_cmd_full	9
dbi_cnd_empty	8
Rsvd	7
gen_d_cmd_busy	6
gen_pld_cfull	5
gen_pld_r_empty	4
gen_pld_w_full	3
gen_pld_w_empty	2
gen_cmd_full	1
gen_cmd_empty	0

Table 5-34 Fields for Register: CMD_PKT_STATUS

Bits	Name	Memory Access	Description
31:28			Reserved Field: Yes
27	dbi_buff_pld_full	R	<p>This bit indicates the full status of the DBI payload internal buffer.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) == 1</p>
26	dbi_buff_pld_empty	R	<p>This bit indicates the empty status of the DBI payload internal buffer.</p> <p>Value After Reset: 0x1</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) == 1</p>
25	dbi_buff_cmd_full	R	<p>This bit indicates the full status of the DBI command internal buffer.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3)) == 1</p>

Bits	Name	Memory Access	Description
24	dbi_buff_cmd_empty	R	<p>This bit indicates the empty status of the DBI command internal buffer.</p> <p>Value After Reset: 0x1</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
23:20			Reserved Field: Yes
19	gen_buff_pld_full	R	<p>This bit indicates the full status of the generic payload internal buffer.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
18	gen_buff_pld_empty	R	<p>This bit indicates the empty status of the generic payload internal buffer.</p> <p>Value After Reset: 0x1</p> <p>Exists: Always</p>
17	gen_buff_cmd_full	R	<p>This bit indicates the full status of the generic command internal buffer.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
16	gen_buff_cmd_empty	R	<p>This bit indicates the empty status of the generic command internal buffer.</p> <p>Value After Reset: 0x1</p> <p>Exists: Always</p>
15			Reserved Field: Yes
14	dbi_rd_cmd_busy	R	<p>This bit is set when a read command is issued and cleared when the entire response is stored in the FIFO for DBI interface.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
13	dbi_pld_r_full	R	<p>This bit indicates the full status of the DBI read payload FIFO.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
12	dbi_pld_r_empty	R	<p>This bit indicates the empty status of the DBI read payload</p> <p>Value After Reset: 0x1</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>

Bits	Name	Memory Access	Description
11	dbi_pld_w_full	R	<p>This bit indicates the full status of the DBI write payload FIFO.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
10	dbi_pld_w_empty	R	<p>This bit indicates the empty status of the DBI write payload FIFO.</p> <p>Value After Reset: 0x1</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
9	dbi_cmd_full	R	<p>This bit indicates the full status of the DBI command FIFO.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
8	dbi_cmd_empy	R	<p>This bit indicates the empty status of the DBI command FIFO.</p> <p>Value After Reset: 0x1</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
7			Reserved Field: Yes
6	gen_rd_cmd_busy	R	<p>This bit is set when a read command is issued and cleared when the entire response is stored in the FIFO for GENERIC interface.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5	gen_pld_r_full	R	<p>This bit indicates the full status of the generic read payload FIFO.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	gen_pld_r_empty	R	<p>This bit indicates the empty status of the generic read payload FIFO.</p> <p>Value After Reset: 0x1</p> <p>Exists: Always</p>
3	gen_pld_w_full	R	<p>This bit indicates the full status of the generic write payload FIFO.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	gen_pld_w_empty	R	<p>This bit indicates the empty status of the generic write payload FIFO.</p> <p>Value After Reset: 0x1</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
1	gen_cmd_full	R	This bit indicates the full status of the generic command FIFO. Value After Reset: 0x0 Exists: Always
0	gen_cmd_empty	R	This bit indicates the empty status of the generic command FIFO. Value After Reset: 0x1 Exists: Always

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.31 TO_CNT_CFG

- **Name:** Timeout Trigger Configuration.
- **Description:** Configures counters that trigger timeout errors. These are used to warn the system of a failure, through an interrupt, and restart the controller in case of unexpected situations that cause deadlock conditions.
- **Size:** 32 bits
- **Offset:** 0x78
- **Exists:** Always

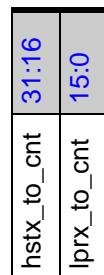


Table 5-35 Fields for Register: TO_CNT_CFG

Bits	Name	Memory Access	Description
31:16	hstx_to_cnt	R/W	<p>This field configures the timeout counter that triggers a high-speed transmission timeout contention detection (measured in TO_CLK_DIVISION cycles).</p> <p>If using non-burst mode and there is not sufficient time to switch from high-speed to low-power and back in the period which is from one line data finishing to the next line sync start, the DSI link returns low-power state once per frame, then you should configure the TO_CLK_DIVISION and hstx_to_cnt to satisfy the following formula: $\text{hstx_to_cnt} * \text{lanebyteclkperiod} * \text{TO_CLK_DIVISION} \geq \text{the time of one FRAME data transmission} * (1 + 10\%)$</p> <p>In burst mode, RGB pixel packets are time-compressed, leaving more time during a scan line. So if in burst mode and there is sufficient time to switch from high-speed to low-power and back in the period of time from one line data finishing to the next line sync start, the DSI link can return low-power mode and back in this time interval to save power. If you choose so, you should configure the TO_CLK_DIVISION and hstx_to_cnt to satisfy the following formula: $\text{hstx_to_cnt} * \text{lanebyteclkperiod} * \text{TO_CLK_DIVISION} \geq \text{the time of one LINE data transmission} * (1 + 10\%)$</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
15:0	lprx_to_cnt	R/W	<p>This field configures the timeout counter that triggers a low-power reception timeout contention detection (measured in TO_CLK_DIVISION cycles).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.32 HS_RD_TO_CNT

- **Name:** Peripheral timeout after HS read operations.
- **Description:** Configures the Peripheral Response timeout after high-speed Read operations.
- **Size:** 32 bits
- **Offset:** 0x7c
- **Exists:** Always

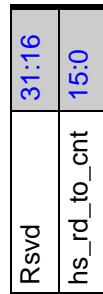


Table 5-36 Fields for Register: HS_RD_TO_CNT

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	hs_rd_to_cnt	R/W	<p>This field sets a period for which DWC_mipi_dsi_host keeps the link still, after sending a high-speed Read operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.33 LP_RD_TO_CNT

- **Name:** Peripheral timeout after LP read operations.
- **Description:** Configures the Peripheral Response timeout after low-power Read operations.
- **Size:** 32 bits
- **Offset:** 0x80
- **Exists:** Always

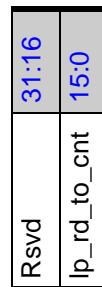


Table 5-37 Fields for Register: LP_RD_TO_CNT

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	lp_rd_to_cnt	R/W	<p>This field sets a period for which DWC_mipi_dsi_host keeps the link still, after sending a low-power Read operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.34 HS_WR_TO_CNT

- **Name:** Peripheral timeout after HS write operations.
- **Description:** Configures the Peripheral Response timeout after high-speed Write operations.
- **Size:** 32 bits
- **Offset:** 0x84
- **Exists:** Always

Rsvd	31:25
	24
Rsvd	23:16
	hs_wr_to_cnt 15:0

Table 5-38 Fields for Register: HS_WR_TO_CNT

Bits	Name	Memory Access	Description
31:25			Reserved Field: Yes
24	presp_to_mode	R/W	When set to 1, this bit causes that peripheral response timeout caused by hs_wr_to_cnt is used only once per eDPI frame, after the following conditions: - dpivsync_edpiwms has risen and fallen. - Packets originated from eDPI have been transmitted and its FIFO is empty again. In this scenario no non-eDPI requests are sent to the D-PHY, even if there is traffic from generic or DBI ready to be sent, making it return to stop state. When it does so, PRESP_TO counter is activated and only when it finishes does the controller send any other traffic that is ready. Value After Reset: 0x0 Exists: (DSI_HOST_DATAINTERFACE == 4)==1
23:16			Reserved Field: Yes
15:0	hs_wr_to_cnt	R/W	This field sets a period for which DWC_mipi_dsi_host keeps the link still, after sending a high-speed Write operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts. Value After Reset: 0x0 Exists: Always

5.1.35 LP_WR_TO_CNT

- **Name:** Peripheral timeout after LP write operations.
- **Description:** Configures the Peripheral Response timeout after low-power Write operations.
- **Size:** 32 bits
- **Offset:** 0x88
- **Exists:** Always

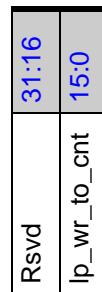


Table 5-39 Fields for Register: LP_WR_TO_CNT

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	lp_wr_to_cnt	R/W	<p>This field sets a period for which DWC_mipi_dsi_host keeps the link still, after sending a low-power Write operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.36 BTA_TO_CNT

- **Name:** Peripheral timeout after BTA completion.
- **Description:** Configures the Peripheral Response timeout after Bus Turnaround completion.
- **Size:** 32 bits
- **Offset:** 0x8c
- **Exists:** Always

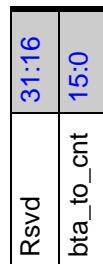


Table 5-40 Fields for Register: BTA_TO_CNT

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	bta_to_cnt	R/W	<p>This field sets a period for which DWC_mipi_dsi_host keeps the link still, after completing a Bus Turnaround. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.37 SDF_3D

- **Name:** 3D information for VSS packets.
- **Description:** Stores 3D control information for VSS packets in video mode.
- **Size:** 32 bits
- **Offset:** 0x90
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

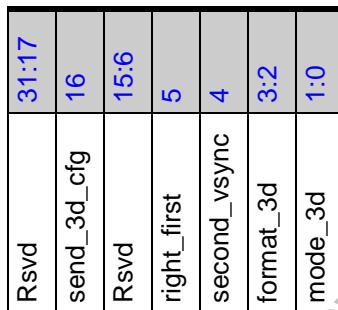


Table 5-41 Fields for Register: SDF_3D

Bits	Name	Memory Access	Description
31:17			Reserved Field: Yes
16	send_3d_cfg	R/W	When set, causes the next VSS packet to include 3D control payload in every VSS packet. Value After Reset: 0x0 Exists: Always
15:6			Reserved Field: Yes
5	right_first	R/W	This bit defines the left/right order: Values: <ul style="list-style-type: none"> ■ 0x0 (LFIRST): left eye is sent first, then right eye ■ 0x1 (RFIRST): right eye data is sent first, then left eye Value After Reset: 0x0 Exists: Always
4	second_vsync	R/W	This field defines whether there is a second VSYNC pulse between Left and Right Images, when 3D Image Format is Frame-based: Values: <ul style="list-style-type: none"> ■ 0x0 (NOSYNC): No sync pulses between left and right data ■ 0x1 (SYNC): Sync pulse HSYNC, VSYNC, blanking between left and right data Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
3:2	format_3d	R/W	<p>This field defines 3D Image Format:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (LINE): Alternating lines of left and right data ■ 0x1 (FRAME): Alternating frames of left and right data ■ 0x2 (PIXEL): Alternating pixels of left and right data ■ 0x3 (RESERVED): Reserved, not used <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1:0	mode_3d	R/W	<p>This field defines 3D Mode On/Off and Display Orientation:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (MODE0): 3D Mode Off , 2D Mode On ■ 0x1 (MODE1): 3D Mode On, Portrait Orientation ■ 0x2 (MODE2): 3D Mode On, Landscape Orientation ■ 0x3 (MODE3): Reserved, not used <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.38 LPCLK_CTRL

- **Name:** Non-continuous Clock configuration.
- **Description:** Configures the possibility for using non continuous clock in the clock lane.
- **Size:** 32 bits
- **Offset:** 0x94
- **Exists:** Always

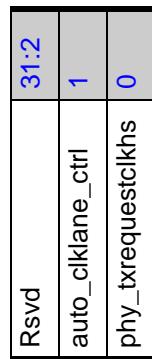


Table 5-42 Fields for Register: LPCLK_CTRL

Bits	Name	Memory Access	Description
31:2			Reserved Field: Yes
1	auto_clklane_ctrl	R/W	<p>This bit enables the automatic mechanism to stop providing clock in the clock lane when time allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
0	phy_txrequestclkhs	R/W	<p>This bit controls the D-PHY PPI txrequestclkhs signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.39 PHY_TMR_LPCLK_CFG

- **Name:** Time configuration for (clock lane) transitions between HS and LP.
- **Description:** Sets the time that DWC_mipi_dsi_host assumes in calculations for the clock lane to switch between high-speed and low-power.
- **Size:** 32 bits
- **Offset:** 0x98
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

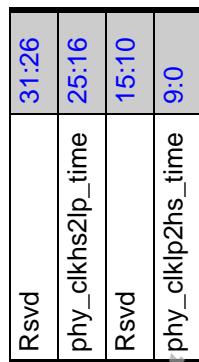


Table 5-43 Fields for Register: PHY_TMR_LPCLK_CFG

Bits	Name	Memory Access	Description
31:26			Reserved Field: Yes
25:16	phy_clkhs2lp_time	R/W	This field configures the maximum time that the D-PHY clock lane takes to go from high-speed to low-power transmission measured in lane byte clock cycles. Value After Reset: 0x0 Exists: Always
15:10			Reserved Field: Yes
9:0	phy_clklp2hs_time	R/W	This field configures the maximum time that the D-PHY clock lane takes to go from low-power to high-speed transmission measured in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.40 PHY_TMR_CFG

- **Name:** Time configuration for (data lanes) transitions between HS and LP.
- **Description:** Sets the time that DWC_mipi_dsi_host assumes in calculations for the data lanes to switch between high-speed and low-power.
- **Size:** 32 bits
- **Offset:** 0x9c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

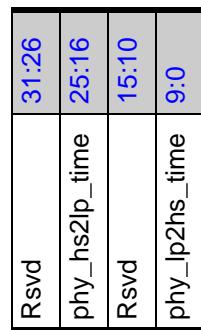


Table 5-44 Fields for Register: PHY_TMR_CFG

Bits	Name	Memory Access	Description
31:26			Reserved Field: Yes
25:16	phy_hs2lp_time	R/W	This field configures the maximum time that the D-PHY data lanes take to go from high-speed to low-power transmission measured in lane byte clock cycles. Value After Reset: 0x0 Exists: Always
15:10			Reserved Field: Yes
9:0	phy_lp2hs_time	R/W	This field configures the maximum time that the D-PHY data lanes take to go from low-power to high-speed transmission measured in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.41 PHY_RSTZ

- **Name:** D-PHY's PLL and Resets.
- **Description:** Controls resets and the PLL of the D-PHY.
- **Size:** 32 bits
- **Offset:** 0xa0
- **Exists:** Always



Table 5-45 Fields for Register: PHY_RSTZ

Bits	Name	Memory Access	Description
31:4			Reserved Field: Yes
3	phy_forcepll	R/W	When the D-PHY is in ULPS, this bit enables the D-PHY PLL. Value After Reset: 0x0 Exists: Always
2	phy_enableclk	R/W	When set to 1, this bit enables the D-PHY Clock Lane Module. Value After Reset: 0x0 Exists: Always
1	phy_rstz	R/W	When set to 0, this bit places the digital section of the D-PHY in the reset state. Value After Reset: 0x0 Exists: Always
0	phy_shutdownz	R/W	When set to 0, this bit places the complete D-PHY macro in power-down state. Value After Reset: 0x0 Exists: Always

5.1.42 PHY_IF_CFG

- **Name:** Active lanes and Stop State minimum time in Stop State.
- **Description:** Configures the number of active lanes and the minimum time to remain in stop state.
- **Size:** 32 bits
- **Offset:** 0xa4
- **Exists:** Always

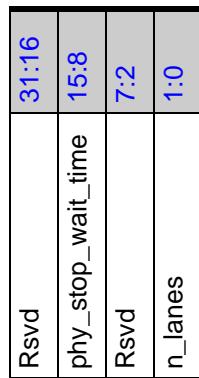


Table 5-46 Fields for Register: PHY_IF_CFG

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:8	phy_stop_wait_time	R/W	This field configures the minimum time PHY needs to stay in StopState before requesting an high-speed transmission Value After Reset: 0x0 Exists: Always
7:2			Reserved Field: Yes
1:0	n_lanes	R/W	This field configures the number of active data lanes: Values: <ul style="list-style-type: none"> ■ 0x0 (ONELANES): lane 0 ■ 0x1 (TWOLANES): lanes 0 and 1 ■ 0x2 (THREELANES): lanes 0, 1, and 2 ■ 0x3 (FOURLANES): lanes 0, 1, 2, and 3 Value After Reset: DSI_HOST_NUMBER_OF_LANES-1 Exists: Always

5.1.43 PHY_ULPS_CTRL

- **Name:** Transitions from and to ULPS, using D-PHY.
- **Description:** Configures entering and leaving ULPS in the D-PHY.
- **Size:** 32 bits
- **Offset:** 0xa8
- **Exists:** Always

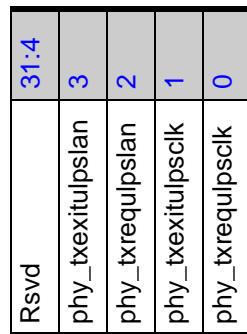


Table 5-47 Fields for Register: PHY_ULPS_CTRL

Bits	Name	Memory Access	Description
31:4			Reserved Field: Yes
3	phy_txexitulpslan	R/W	ULPS mode Exit on all active data lanes. Value After Reset: 0x0 Exists: Always
2	phy_txrequlpslan	R/W	ULPS mode Request on all active data lanes. Value After Reset: 0x0 Exists: Always
1	phy_txexitulpsclk	R/W	ULPS mode Exit on clock lane. Value After Reset: 0x0 Exists: Always
0	phy_txrequlpsclk	R/W	ULPS mode Request on clock lane. Value After Reset: 0x0 Exists: Always

5.1.44 PHY_TX_TRIGGER

- **Name:** Pins related to D-PHY triggers.
- **Description:** Configures the pins that activate triggers in the D-PHY.
- **Size:** 32 bits
- **Offset:** 0xac
- **Exists:** Always

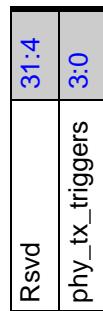


Table 5-48 Fields for Register: PHY_TX_TRIGGER

Bits	Name	Memory Access	Description
31:4			Reserved Field: Yes
3:0	phy_tx_triggers	R/W	This field controls the trigger transmissions. Value After Reset: 0x0 Exists: Always

5.1.45 PHY_STATUS

- **Name:** D-PHY Status.
- **Description:** Contains information about the status of the D-PHY.
- **Size:** 32 bits
- **Offset:** 0xb0
- **Exists:** Always

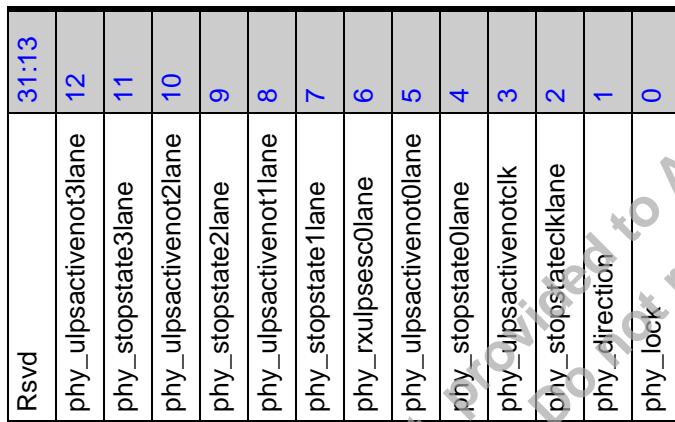


Table 5-49 Fields for Register: PHY_STATUS

Bits	Name	Memory Access	Description
31:13			Reserved Field: Yes
12	phy_ulpsactivenot3lane	R	<p>This bit indicates the status of ulpsactivenot3lane D-PHY signal.</p> <p>Value After Reset: * Varies</p> <p>Exists: DSI_HOST_NUMBER_OF_LANES>3</p> <p>Testable: untestable</p> <p>Volatile: true</p>
11	phy_stopstate3lane	R	<p>This bit indicates the status of phystopstate3lane D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_NUMBER_OF_LANES>3</p>
10	phy_ulpsactivenot2lane	R	<p>This bit indicates the status of ulpsactivenot2lane D-PHY signal.</p> <p>Value After Reset: * Varies</p> <p>Exists: DSI_HOST_NUMBER_OF_LANES>2</p> <p>Testable: untestable</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
9	phy_stopstate2lane	R	<p>This bit indicates the status of phystopstate2lane D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_NUMBER_OF_LANES>2</p>
8	phy_ulpssactivenot1lane	R	<p>This bit indicates the status of ulpsactivenot1lane D-PHY signal.</p> <p>Value After Reset: * Varies</p> <p>Exists: DSI_HOST_NUMBER_OF_LANES>1</p> <p>Testable: untestable</p> <p>Volatile: true</p>
7	phy_stopstate1lane	R	<p>This bit indicates the status of phystopstate1lane D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_NUMBER_OF_LANES>1</p>
6	phy_rxulpsesc0lane	R	<p>This bit indicates the status of rxulpsesc0lane D-PHY signal.</p> <p>Value After Reset: 0x1</p> <p>Exists: Always</p>
5	phy_ulpssactivenot0lane	R	<p>This bit indicates the status of ulpsactivenot0lane D-PHY signal.</p> <p>Value After Reset: * Varies</p> <p>Exists: Always</p> <p>Testable: untestable</p> <p>Volatile: true</p>
4	phy_stopstate0lane	R	<p>This bit indicates the status of phystopstate0lane D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	phy_ulpssactivenotclk	R	<p>This bit indicates the status of phyulpssactivenotclk D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	phy_stopstateclklane	R	<p>This bit indicates the status of phystopstateclklane D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1	phy_direction	R	<p>This bit indicates the status of phydirection D-PHY signal.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
0	phy_lock	R	This bit indicates the status of phylock D-PHY signal. Value After Reset: 0x0 Exists: Always

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.46 PHY_TST_CTRL0

- **Name:** D-PHY control and clear pins.
- **Description:** Controls clock and clear pins of the D-PHY vendor specific interface.
- **Size:** 32 bits
- **Offset:** 0xb4
- **Exists:** Always

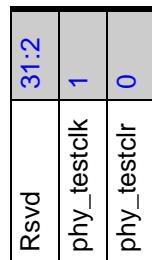


Table 5-50 Fields for Register: PHY_TST_CTRL0

Bits	Name	Memory Access	Description
31:2			Reserved Field: Yes
1	phy_testclk	R/W	This bit is used to clock the TESTDIN bus into the D-PHY. Value After Reset: 0x0 Exists: Always
0	phy_testclr	R/W	PHY test interface clear (active high). Value After Reset: 0x1 Exists: Always

5.1.47 PHY_TST_CTRL1

- **Name:** D-PHY data and enable pins.
- **Description:** Controls data and enable pins of the D-PHY vendor specific interface.
- **Size:** 32 bits
- **Offset:** 0xb8
- **Exists:** Always

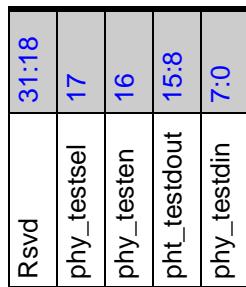


Table 5-51 Fields for Register: PHY_TST_CTRL1

Bits	Name	Memory Access	Description
31:18			Reserved Field: Yes
17	phy_testsel	R/W	PHY BiDIR MUX selector Value After Reset: 0x0 Exists: Always
16	phy_testen	R/W	PHY test interface operation selector: Values: <ul style="list-style-type: none"> ■ 0x1 (WRITEADDR): the address write operation is set on the falling edge of the testclk signal. ■ 0x0 (WRITEDATA): the data write operation is set on the rising edge of the testclk signal. Value After Reset: 0x0 Exists: Always
15:8	pht_testdout	R	PHY output 8-bit data bus for read-back and internal probing functionalities. Value After Reset: 0x0 Exists: Always
7:0	phy_testdin	R/W	PHY test interface input 8-bit data bus for internal register programming and test functionalities access. Value After Reset: 0x0 Exists: Always

5.1.48 INT_ST0

- **Name:** Interrupts status 0.
- **Description:** Contains the status of interrupt sources from acknowledge reports and the D-PHY.
- **Size:** 32 bits
- **Offset:** 0xbc
- **Exists:** Always

Rsvd	31:21
dphy_errors_4	20
dphy_errors_3	19
dphy_errors_2	18
dphy_errors_1	17
dphy_errors_0	16
ack_with_err_15	15
ack_with_err_14	14
ack_with_err_13	13
ack_with_err_12	12
ack_with_err_11	11
ack_with_err_10	10
ack_with_err_9	9
ack_with_err_8	8
ack_with_err_7	7
ack_with_err_6	6
ack_with_err_5	5
ack_with_err_4	4
ack_with_err_3	3
ack_with_err_2	2
ack_with_err_1	1
ack_with_err_0	0

Table 5-52 Fields for Register: INT_ST0

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	dphy_errors_4	RC	This bit indicates LP1 contention error ErrContentionLP1 from Lane 0. Value After Reset: 0x0 Exists: Always
19	dphy_errors_3	RC	This bit indicates LP0 contention error ErrContentionLP0 from Lane 0. Value After Reset: 0x0 Exists: Always
18	dphy_errors_2	RC	This bit indicates control error ErrControl from Lane 0. Value After Reset: 0x0 Exists: Always
17	dphy_errors_1	RC	This bit indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0. Value After Reset: 0x0 Exists: Always
16	dphy_errors_0	RC	This bit indicates ErrEsc escape entry error from Lane 0. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15	ack_with_err_15	RC	<p>This bit retrieves the DSI protocol violation from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
14	ack_with_err_14	RC	<p>This bit retrieves the reserved (specific to device) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	ack_with_err_13	RC	<p>This bit retrieves the invalid transmission length from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
12	ack_with_err_12	RC	<p>This bit retrieves the DSI VC ID Invalid from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	ack_with_err_11	RC	<p>This bit retrieves the not recognized DSI data type from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	ack_with_err_10	RC	<p>This bit retrieves the checksum error (long packet only) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	ack_with_err_9	RC	<p>This bit retrieves the ECC error, multi-bit (detected, not corrected) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	ack_with_err_8	RC	<p>This bit retrieves the ECC error, single-bit (detected and corrected) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	ack_with_err_7	RC	<p>This bit retrieves the Contention Detected error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
6	ack_with_err_6	RC	<p>This bit retrieves the False Control error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5	ack_with_err_5	RC	<p>This bit retrieves the Peripheral Timeout error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	ack_with_err_4	RC	<p>This bit retrieves the low-power Transmit Sync error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	ack_with_err_3	RC	<p>This bit retrieves the Escape Mode Entry Command error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	ack_with_err_2	RC	<p>This bit retrieves the EoT Sync error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1	ack_with_err_1	RC	<p>This bit retrieves the SoT Sync error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
0	ack_with_err_0	RC	<p>This bit retrieves the SoT error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.49 INT_ST1

- **Name:** Interrupts Status 1.
- **Description:** Contains the status of interrupt sources related to timeouts, ECC, CRC, packet size, EoTp, Generic and DBI interfaces.
- **Size:** 32 bits
- **Offset:** 0xc0
- **Exists:** Always

Rsvd	31:21
tear_request_err	20
dpi_buff_pld_under	19
Rsvd	18
dbi_illegal_comm_err	17
dbi_pld_recv_err	16
dbi_pld_rd_err	15
dbi_pld_wr_err	14
dbi_cmd_wr_err	13
gen_pld_recev_err	12
gen_pld_rd_err	11
gen_pld_send_err	10
gen_pld_wr_err	9
gen_cmd_wr_err	8
dpi_pld_wr_err	7
eopt_eri	6
pkt_size_eri	5
crc_eri	4
ecc_multpl_err	3
ecc_single_err	2
to_lp_rx	1
to_hs_tx	0

Table 5-53 Fields for Register: INT_ST1

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	tear_request_err	RC	This bit indicates that tear_request has occurred but tear effect is not active in DSI host and device. Value After Reset: 0x0 Exists: (DSI_HOST_DATAINTERFACE == 4)==1
19	dpi_buff_pld_under	RC	This bit indicates that an underflow has occurred when reading payload to build DSI packet for video mode Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
18			Reserved Field: Yes
17	dbi_illegal_comm_err	RC	This bit indicates that an attempt to write an illegal command on the DBI interface is made and the controller is blocked by transmission. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1

Bits	Name	Memory Access	Description
16	dbi_pld_recv_err	RC	<p>This bit indicates that during a DBI read back packet, the payload FIFO becomes full and the received data is corrupted.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
15	dbi_pld_rd_err	RC	<p>This bit indicates that during a DCS read data, the payload FIFO goes empty and the data sent to the interface is corrupted.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
14	dbi_pld_wr_err	RC	<p>This bit indicates that the system tried to write the payload data through the DBI interface and the FIFO is full. Therefore, the command is not written.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
13	dbi_cmd_wr_err	RC	<p>This bit indicates that the system tried to write a command through the DBI but the command FIFO is full. Therefore, the command is not written.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
12	gen_pld_recev_err	RC	<p>This bit indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	gen_pld_rd_err	RC	<p>This bit indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	gen_pld_send_err	RC	<p>This bit indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	gen_pld_wr_err	RC	<p>This bit indicates that the system tried to write a payload data through the Generic interface and the FIFO is full. Therefore, the payload is not written.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
8	gen_cmd_wr_err	RC	<p>This bit indicates that the system tried to write a command through the Generic interface and the FIFO is full. Therefore, the command is not written.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	dpi_pld_wr_err	RC	<p>This bit indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1</p>
6	eotp_err	RC	<p>This bit indicates that the EoTp packet has not been received at the end of the incoming peripheral transmission.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5	pkt_size_err	RC	<p>This bit indicates that the packet size error has been detected during the packet reception.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	crc_err	RC	<p>This bit indicates that the CRC error has been detected in the received packet payload.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	ecc_multpl_err	RC	<p>This bit indicates that the ECC multiple error has been detected in a received packet.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	ecc_single_err	RC	<p>This bit indicates that the ECC single error has been detected and corrected in a received packet.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1	to_lp_rx	RC	<p>This bit indicates that the low-power reception timeout counter reached the end and contention has been detected.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
0	to_hs_tx	RC	<p>This bit indicates that the high-speed transmission timeout counter reached the end and contention has been detected.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.50 INT_MSK0

- **Name:** INT_ST0 mask.
- **Description:** Configures masks for the sources of interrupts that affect the INT_ST0 register. Write 1 to un-mask each error report.
- **Size:** 32 bits
- **Offset:** 0xc4
- **Exists:** Always

Rsvd	31:21
mask_dphy_errors_4	20
mask_dphy_errors_3	19
mask_dphy_errors_2	18
mask_dphy_errors_1	17
mask_dphy_errors_0	16
mask_ack_with_err_15	15
mask_ack_with_err_14	14
mask_ack_with_err_13	13
mask_ack_with_err_12	12
mask_ack_with_err_11	11
mask_ack_with_err_10	10
mask_ack_with_err_9	9
mask_ack_with_err_8	8
mask_ack_with_err_7	7
mask_ack_with_err_6	6
mask_ack_with_err_5	5
mask_ack_with_err_4	4
mask_ack_with_err_3	3
mask_ack_with_err_2	2
mask_ack_with_err_1	1
mask_ack_with_err_0	0

Table 5-54 Fields for Register: INT_MSK0

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	mask_dphy_errors_4	R/W	Mask for dphy_errors_4 Value After Reset: 0x0 Exists: Always
19	mask_dphy_errors_3	R/W	Mask for dphy_errors_3 Value After Reset: 0x0 Exists: Always
18	mask_dphy_errors_2	R/W	Mask for dphy_errors_2 Value After Reset: 0x0 Exists: Always
17	mask_dphy_errors_1	R/W	Mask for dphy_errors_1 Value After Reset: 0x0 Exists: Always
16	mask_dphy_errors_0	R/W	Mask for dphy_errors_0 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15	mask_ack_with_err_15	R/W	Mask for ack_with_err_15 Value After Reset: 0x0 Exists: Always
14	mask_ack_with_err_14	R/W	Mask for ack_with_err_14 Value After Reset: 0x0 Exists: Always
13	mask_ack_with_err_13	R/W	Mask for ack_with_err_13 Value After Reset: 0x0 Exists: Always
12	mask_ack_with_err_12	R/W	Mask for ack_with_err_12 Value After Reset: 0x0 Exists: Always
11	mask_ack_with_err_11	R/W	Mask for ack_with_err_11 Value After Reset: 0x0 Exists: Always
10	mask_ack_with_err_10	R/W	Mask for ack_with_err_10 Value After Reset: 0x0 Exists: Always
9	mask_ack_with_err_9	R/W	Mask for ack_with_err_9 Value After Reset: 0x0 Exists: Always
8	mask_ack_with_err_8	R/W	Mask for ack_with_err_8 Value After Reset: 0x0 Exists: Always
7	mask_ack_with_err_7	R/W	Mask for ack_with_err_7 Value After Reset: 0x0 Exists: Always
6	mask_ack_with_err_6	R/W	Mask for ack_with_err_6 Value After Reset: 0x0 Exists: Always
5	mask_ack_with_err_5	R/W	Mask for ack_with_err_5 Value After Reset: 0x0 Exists: Always
4	mask_ack_with_err_4	R/W	Mask for ack_with_err_4 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
3	mask_ack_with_err_3	R/W	Mask for ack_with_err_3 Value After Reset: 0x0 Exists: Always
2	mask_ack_with_err_2	R/W	Mask for ack_with_err_2 Value After Reset: 0x0 Exists: Always
1	mask_ack_with_err_1	R/W	Mask for ack_with_err_1 Value After Reset: 0x0 Exists: Always
0	mask_ack_with_err_0	R/W	Mask for ack_with_err_0 Value After Reset: 0x0 Exists: Always

5.1.51 INT_MSK1

- **Name:** INT_ST1 mask.
- **Description:** Configures masks for the sources of interrupts that affect the INT_ST1 register.
- **Size:** 32 bits
- **Offset:** 0xc8
- **Exists:** Always

	31:21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsvd																						
mask_tear_request_err																						
mask_dpi_buff_pld_under																						
Rsvd																						
mask_dbi_illegal_comm_err																						
mask_dbi_pld_recv_err																						
mask_dbi_pld_rd_err																						
mask_dbi_pld_wr_err																						
mask_gen_pld_recv_err																						
mask_gen_pld_rd_err																						
mask_gen_pld_send_err																						
mask_gen_pld_wr_err																						
mask_gen_cmd_wr_err																						
mask_dpi_pld_wr_err																						
mask_eopt_err																						
mask_pkt_size_err																						
mask_circ_err																						
mask_ecc_mullpi_err																						
mask_ecc_single_err																						
mask_to_lp_rx																						
mask_to_hs_tx																						

Table 5-55 Fields for Register: INT_MSK1

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	mask_tear_request_err	R/W	Mask for tear_request_err Value After Reset: 0x0 Exists: (DSI_HOST_DATAINTERFACE == 4)==1
19	mask_dpi_buff_pld_under	R/W	Mask for dpi_buff_pld_under Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
18			Reserved Field: Yes
17	mask_dbi_illegal_comm_err	R/W	Mask for dbi_illegal_comm_err Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1

Bits	Name	Memory Access	Description
16	mask_dbi_pld_recv_err	R/W	<p>Mask for dbi_pld_recv_err</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
15	mask_dbi_pld_rd_err	R/W	<p>Mask for dbi_pld_rd_err</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
14	mask_dbi_pld_wr_err	R/W	<p>Mask for dbi_pld_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
13	mask_dbi_cmd_wr_err	R/W	<p>Mask for dbi_cmd_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
12	mask_gen_pld_recev_err	R/W	<p>Mask for gen_pld_recev_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	mask_gen_pld_rd_err	R/W	<p>Mask for gen_pld_rd_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	mask_gen_pld_send_err	R/W	<p>Mask for gen_pld_send_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	mask_gen_pld_wr_err	R/W	<p>Mask for gen_pld_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	mask_gen_cmd_wr_err	R/W	<p>Mask for gen_cmd_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	mask_dpi_pld_wr_err	R/W	<p>Mask for dpi_pld_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1</p>

Bits	Name	Memory Access	Description
6	mask_eopt_err	R/W	Mask for eopt_err Value After Reset: 0x0 Exists: Always
5	mask_pkt_size_err	R/W	Mask for pkt_size_err Value After Reset: 0x0 Exists: Always
4	mask_crc_err	R/W	Mask for crc_err Value After Reset: 0x0 Exists: Always
3	mask_ecc_multpl_err	R/W	Mask for ecc_multpl_err Value After Reset: 0x0 Exists: Always
2	mask_ecc_single_err	R/W	Mask for ecc_single_err Value After Reset: 0x0 Exists: Always
1	mask_to_lp_rx	R/W	Mask for to_lp_rx Value After Reset: 0x0 Exists: Always
0	mask_to_hs_tx	R/W	Mask for to_hs_tx Value After Reset: 0x0 Exists: Always

5.1.52 PHY_CAL

- **Name:** D-PHY skew calibration.
- **Description:** Controls the skew calibration of D-PHY.
- **Size:** 32 bits
- **Offset:** 0xcc
- **Exists:** (DSI_HOST_GEN31PHY == 1) || (DSI_HOST_GEN32PHY == 1) || (DSI_HOST_SNPS_PHY == 0)==1

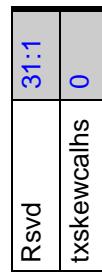


Table 5-56 Fields for Register: PHY_CAL

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	txskewcalhs	R/W	High-speed skew calibration is started when txskewcalhs is set high (assuming that PHY is in Stop state). Value After Reset: 0x0 Exists: Always

5.1.53 INT_FORCE0

- **Name:** Force INT_ST0
- **Description:** Forces interrupt that affect the INT_ST0 register.
- **Size:** 32 bits
- **Offset:** 0xd8
- **Exists:** Always

Rsvd	31:21
force_dphy_errors_4	20
force_dphy_errors_3	19
force_dphy_errors_2	18
force_dphy_errors_1	17
force_dphy_errors_0	16
force_ack_with_err_15	15
force_ack_with_err_14	14
force_ack_with_err_13	13
force_ack_with_err_12	12
force_ack_with_err_11	11
force_ack_with_err_10	10
force_ack_with_err_9	9
force_ack_with_err_8	8
force_ack_with_err_7	7
force_ack_with_err_6	6
force_ack_with_err_5	5
force_ack_with_err_4	4
force_ack_with_err_3	3
force_ack_with_err_2	2
force_ack_with_err_1	1
force_ack_with_err_0	0

Table 5-57 Fields for Register: INT_FORCE0

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	force_dphy_errors_4	R/W	Force dphy_errors_4 Value After Reset: 0x0 Exists: Always
19	force_dphy_errors_3	R/W	Force dphy_errors_3 Value After Reset: 0x0 Exists: Always
18	force_dphy_errors_2	R/W	Force dphy_errors_2 Value After Reset: 0x0 Exists: Always
17	force_dphy_errors_1	R/W	Force dphy_errors_1 Value After Reset: 0x0 Exists: Always
16	force_dphy_errors_0	R/W	Force dphy_errors_0 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15	force_ack_with_err_15	R/W	Force ack_with_err_15 Value After Reset: 0x0 Exists: Always
14	force_ack_with_err_14	R/W	Force ack_with_err_14 Value After Reset: 0x0 Exists: Always
13	force_ack_with_err_13	R/W	Force ack_with_err_13 Value After Reset: 0x0 Exists: Always
12	force_ack_with_err_12	R/W	Force ack_with_err_12 Value After Reset: 0x0 Exists: Always
11	force_ack_with_err_11	R/W	Force ack_with_err_11 Value After Reset: 0x0 Exists: Always
10	force_ack_with_err_10	R/W	Force ack_with_err_10 Value After Reset: 0x0 Exists: Always
9	force_ack_with_err_9	R/W	Force ack_with_err_9 Value After Reset: 0x0 Exists: Always
8	force_ack_with_err_8	R/W	Force ack_with_err_8 Value After Reset: 0x0 Exists: Always
7	force_ack_with_err_7	R/W	Force ack_with_err_7 Value After Reset: 0x0 Exists: Always
6	force_ack_with_err_6	R/W	Force ack_with_err_6 Value After Reset: 0x0 Exists: Always
5	force_ack_with_err_5	R/W	Force ack_with_err_5 Value After Reset: 0x0 Exists: Always
4	force_ack_with_err_4	R/W	Force ack_with_err_4 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
3	force_ack_with_err_3	R/W	Force ack_with_err_3 Value After Reset: 0x0 Exists: Always
2	force_ack_with_err_2	R/W	Force ack_with_err_2 Value After Reset: 0x0 Exists: Always
1	force_ack_with_err_1	R/W	Force ack_with_err_1 Value After Reset: 0x0 Exists: Always
0	force_ack_with_err_0	R/W	Force ack_with_err_0 Value After Reset: 0x0 Exists: Always

5.1.54 INT_FORCE1

- **Name:** Force INT_ST1.
- **Description:** Forces interrupts that affect the INT_ST1 register.
- **Size:** 32 bits
- **Offset:** 0xdc
- **Exists:** Always

Rsvd	31:21
force_tear_request_err	20
force_dpi_buff_pld_under	19
Rsvd	18
force_db_ilegal_comm_err	17
force_db_ipld_recv_err	16
force_db_ipld_rd_err	15
force_db_ipld_wr_err	14
force_db_icmd_wr_err	13
force_gen_pld_recv_err	12
force_gen_pld_rd_err	11
force_gen_pld_send_err	10
force_gen_pld_wr_err	9
force_gen_icmd_wr_err	8
force_qpi_pld_wr_err	7
force_eopt_err	6
force_pkt_size_err	5
force_crc_err	4
force_ecc_multip_err	3
force_ecc_single_err	2
force_to_lp_rx	1
force_to_hs_tx	0

Table 5-58 Fields for Register: INT_FORCE1

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	force_tear_request_err	R/W	Force for tear_request_err. Value After Reset: 0x0 Exists: (DSI_HOST_DATAINTERFACE == 4)==1
19	force_dpi_buff_pld_under	R/W	Force for dpi_buff_pld_under. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
18			Reserved Field: Yes
17	force_db_ilegal_comm_err	R/W	Force dbi_illegal_comm_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1

Bits	Name	Memory Access	Description
16	force_dbi_pld_recv_err	R/W	<p>Force dbi_pld_recv_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
15	force_dbi_pld_rd_err	R/W	<p>Force dbi_pld_rd_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
14	force_dbi_pld_wr_err	R/W	<p>Force dbi_pld_wr_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
13	force_dbi_cmd_wr_err	R/W	<p>Force dbi_cmd_wr_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))==1</p>
12	force_gen_pld_recev_err	R/W	<p>Force gen_pld_recev_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	force_gen_pld_rd_err	R/W	<p>Force gen_pld_rd_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	force_gen_pld_send_err	R/W	<p>Force gen_pld_send_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	force_gen_pld_wr_err	R/W	<p>Force gen_pld_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	force_gen_cmd_wr_err	R/W	<p>Force gen_cmd_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	force_dpi_pld_wr_err	R/W	<p>Force dpi_pld_wr_err</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1</p>

Bits	Name	Memory Access	Description
6	force_eopt_err	R/W	Force eopt_err Value After Reset: 0x0 Exists: Always
5	force_pkt_size_err	R/W	Force pkt_size_err Value After Reset: 0x0 Exists: Always
4	force_crc_err	R/W	Force crc_err Value After Reset: 0x0 Exists: Always
3	force_ecc_multpl_err	R/W	Force ecc_multpl_err Value After Reset: 0x0 Exists: Always
2	force_ecc_single_err	R/W	Force ecc_single_err Value After Reset: 0x0 Exists: Always
1	force_to_lp_rx	R/W	Force to_lp_rx Value After Reset: 0x0 Exists: Always
0	force_to_hs_tx	R/W	Force to_hs_tx Value After Reset: 0x0 Exists: Always

5.1.55 AUTO_ULPS_MODE

- **Name:** Automatic ULPS control.
- **Description:** Configures automatic ULPS control.
- **Size:** 32 bits
- **Offset:** 0xe0
- **Exists:** (DSI_HOST_DATAINTERFACE == 4)==1

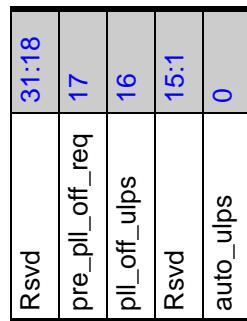


Table 5-59 Fields for Register: AUTO_ULPS_MODE

Bits	Name	Memory Access	Description
31:18			Reserved Field: Yes
17	pre_pll_off_req	R/W	When pll_off_ulps is active, allows to turn off PLL before the request to enter in ULPS. Value After Reset: 0x0 Exists: Always
16	pll_off_ulps	R/W	Turn off the D-PHY PLL during ULPS Value After Reset: 0x0 Exists: Always
15:1			Reserved Field: Yes
0	auto_ulps	R/W	This bit enables the automatic mechanism to enter and exit ULPS Value After Reset: 0x0 Exists: Always

5.1.56 AUTO_ULPS_ENTRY_DELAY

- **Name:** ULPS transition delay.
- **Description:** Configures the delay (in lanebyteclk) to wait before entering ULPS.
- **Size:** 32 bits
- **Offset:** 0xe4
- **Exists:** (DSI_HOST_DATAINTERFACE == 4)==1



Table 5-60 Fields for Register: AUTO_ULPS_ENTRY_DELAY

Bits	Name	Memory Access	Description
31:0	ulps_entry_delay	R/W	<p>Configures the delay (in lanebyteclk) to wait before entering ULPS</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.57 AUTO_ULPS_WAKEUP_TIME

- **Name:** D-PHY wakeup time.
- **Description:** Configures the DPHY wakeup time (in pclk).
- **Size:** 32 bits
- **Offset:** 0xe8
- **Exists:** (DSI_HOST_DATAINTERFACE == 4)==1



Table 5-61 Fields for Register: AUTO_ULPS_WAKEUP_TIME

Bits	Name	Memory Access	Description
31:16	twakeup_cnt	R/W	Twakeup counter Value After Reset: 0x0 Exists: Always
15:0	twakeup_clk_div	R/W	Twakeup clock divider Value After Reset: 0x0 Exists: Always

5.1.58 DSC_PARAMETER

- **Name:** Display Stream Compression.
- **Description:** Configures Display Stream Compression.
- **Size:** 32 bits
- **Offset:** 0xf0
- **Exists:** Always

Rsvd	31:18
pps_sel	17:16
Rsvd	15:10
compress_algo	9:8
Rsvd	7:1
compression_mode	0

Table 5-62 Fields for Register: DSC_PARAMETER

Bits	Name	Memory Access	Description
31:18			Reserved Field: Yes
17:16	pps_sel	R/W	<p>This field indicates the PPS selector: Values:</p> <ul style="list-style-type: none"> ■ 0x0 (PPSSEL0): PPS Table 1 ■ 0x1 (PPSSEL1): PPS Table 2 ■ 0x2 (PPSSEL2): PPS Table 3 ■ 0x3 (PPSSEL3): PPS Table 4 <p>Value After Reset: 0x0 Exists: Always</p>
15:10			Reserved Field: Yes
9:8	compress_algo	R/W	<p>This field indicates the algorithm identifier: Values:</p> <ul style="list-style-type: none"> ■ 0x0 (ALGO0): VESA DSC Standard 1.1 ■ 0x1 (ALGO1): reserved, not used ■ 0x2 (ALGO2): reserved, not used ■ 0x3 (ALGO3): vendor-specific algorithm <p>Value After Reset: 0x0 Exists: Always</p>
7:1			Reserved Field: Yes

Bits	Name	Memory Access	Description
0	compression_mode	R/W	When set to 1, this bit enables the compression mode. Value After Reset: 0x0 Exists: Always

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.59 PHY_TMR_RD_CFG

- **Name:** PHY timings.
- **Description:** Configures times related to PHY to perform some operations in lane byte clock cycles.
- **Size:** 32 bits
- **Offset:** 0xf4
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

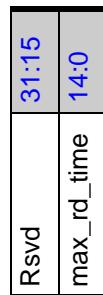


Table 5-63 Fields for Register: PHY_TMR_RD_CFG

Bits	Name	Memory Access	Description
31:15			Reserved Field: Yes
14:0	max_rd_time	R/W	<p>This field configures the maximum time required to perform a read command in lane byte clock cycles. This register can only be modified when no read command is in progress.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.60 AUTO_ULPS_MIN_TIME

- **Name:** PHY Timings - Transition between ulpsactivenot and ulpsexitreq.
- **Description:** Configures the minimum time required by PHY between ulpsactivenot and ulpsexitreq for clock and data lane.
- **Size:** 32 bits
- **Offset:** 0xf8
- **Exists:** (DSI_HOST_DATAINTERFACE == 4)==1

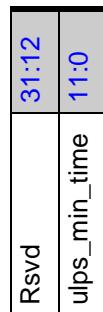


Table 5-64 Fields for Register: AUTO_ULPS_MIN_TIME

Bits	Name	Memory Access	Description
31:12			Reserved Field: Yes
11:0	ulps_min_time	R/W	Configures the minimum time required by PHY between ulpsactivenot and ulpsexitreq for clock and data lane. Value After Reset: 0x0 Exists: Always

5.1.61 PHY_MODE

- **Name:** PHY interface.
- **Description:** Select the PHY interface.
- **Size:** 32 bits
- **Offset:** 0xfc
- **Exists:** DSI_HOST_PHY==2

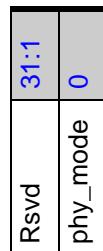


Table 5-65 Fields for Register: PHY_MODE

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	phy_mode	R/W	<p>This bit is to select the PHY interface to be used:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (DPHY): Select DPHY ■ 0x1 (CPHY): Select CPHY <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.62 VID_SHADOW_CTRL

- **Name:** DPI Shadow Feature.
- **Description:** Controls dpi shadow feature
- **Size:** 32 bits
- **Offset:** 0x100
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1



Table 5-66 Fields for Register: VID_SHADOW_CTRL

Bits	Name	Memory Access	Description
31:17			Reserved Field: Yes
16	vid_shadow_pin_req	R/W	When set to 1, the video request is done by external pin. In this mode vid_shadow_req is ignored. Value After Reset: 0x0 Exists: Always
15:9			Reserved Field: Yes
8	vid_shadow_req	R/W	When set to 1, this bit request that the dpi registers from regbank are copied to the auxiliary registers. When the request is completed this bit is auto clear. Value After Reset: 0x0 Exists: Always
7:1			Reserved Field: Yes
0	vid_shadow_en	R/W	When set to 1, DPI receives the active configuration from the auxiliary registers. When the feature is set at the same time than vid_shadow_req the auxiliary registers are automatically updated. Value After Reset: 0x0 Exists: Always

5.1.63 DPI_VCID_ACT

- **Name:** Actual DPI Virtual Channel ID.
- **Description:** Holds the value that controller is using for DPI_VCID.
- **Size:** 32 bits
- **Offset:** 0x10c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

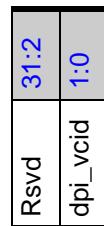


Table 5-67 Fields for Register: DPI_VCID_ACT

Bits	Name	Memory Access	Description
31:2			Reserved Field: Yes
1:0	dpi_vcid	R	This field specifies the DPI virtual channel id that is indexed to the Video mode packets. Value After Reset: 0x0 Exists: Always

5.1.64 DPI_COLOR_CODING_ACT

- **Name:** Actual DPI Color Coding
- **Description:** Holds the value that controller is using for DPI_COLOR_CODING.
- **Size:** 32 bits
- **Offset:** 0x110
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

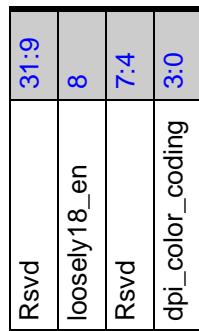


Table 5-68 Fields for Register: DPI_COLOR_CODING_ACT

Bits	Name	Memory Access	Description
31:9			Reserved Field: Yes
8	loosely18_en	R	When 1, this bit activates loosely packed variant to 18-bit configurations. Value After Reset: 0x0 Exists: Always
7:4			Reserved Field: Yes

Bits	Name	Memory Access	Description
3:0	dpi_color_coding	R	<p>This field configures the DPI color for Video Mode/eDPI Command Mode coding as follows:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (CC00): 16-bit configuration 1 ■ 0x1 (CC01): 16-bit configuration 2 ■ 0x2 (CC02): 16-bit configuration 3 ■ 0x3 (CC03): 18-bit configuration 1 ■ 0x4 (CC04): 18-bit configuration 2 ■ 0x5 (CC05): 24-bit ■ 0x6 (CC06): 20-bit YCbCr 4:2:2 loosely packed / Reserved for eDPI Command Mode ■ 0x7 (CC07): 24-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode ■ 0x8 (CC08): 16-bit YCbCr 4:2:2 / Reserved for eDPI Command Mode ■ 0x9 (CC09): 30-bit - DSC_ENC 10bit / Reserved for eDPI Command Mode ■ 0xa (CC10): 36-bit / Reserved for eDPI Command Mode ■ 0xb (CC11): 12-bit YCbCr 4:2:0 / Reserved for eDPI Command Mode ■ 0xc (CC12): DSC24 compressed Data ■ 0xd (CC13): Reserved ■ 0xe (CC14): Reserved ■ 0xf (CC15): Reserved <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.65 DPI_LP_CMD_TIM_ACT

- **Name:** Actual DPI Low Power Commands' Timing.
- **Description:** Holds the value that controller is using for DPI_LP_CMD_TIM.
- **Size:** 32 bits
- **Offset:** 0x118
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

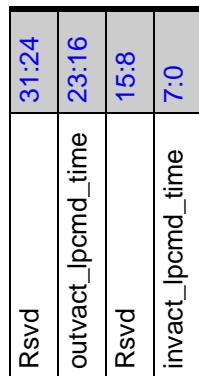


Table 5-69 Fields for Register: DPI_LP_CMD_TIM_ACT

Bits	Name	Memory Access	Description
31:24			Reserved Field: Yes
23:16	outvact_lpcmd_time		This field is used for the transmission of commands in low-power mode. It specifies the size, in bytes, of the largest packet that can fit in a line during the VSA, VBP, and VFP regions. Value After Reset: 0x0 Exists: Always
15:8			Reserved Field: Yes
7:0	invact_lpcmd_time	R	This field is used for the transmission of commands in low-power mode. It specifies the size, in bytes, of the largest packet that can fit in a line during the VACT region. Value After Reset: 0x0 Exists: Always

5.1.66 EDPI_TE_HW_CFG

- **Name:** TE for Hardware operations.
- **Description:** Configures the tearing effect by Hardware operations.
- **Size:** 32 bits
- **Offset:** 0x11c
- **Exists:** (DSI_HOST_DATAINTERFACE == 4)==1

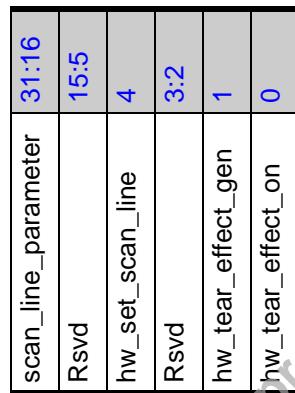


Table 5-70 Fields for Register: EDPI_TE_HW_CFG

Bits	Name	Memory Access	Description
31:16	scan_line_parameter	R/W	<p>When hw_set_scan_line = 1, this bit configures the parameter that describes the Tearing Effect Output Line mode.</p> <p>Scan_line_parameter[31:24] corresponds to Parameter 1 and Scan_line_parameter[23:16] corresponds to Parameter 2.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:5			Reserved Field: Yes
4	hw_set_scan_line	R/W	<p>When hw_tear_effect_on = 1, this bit configures DCS packet type to be issued by DWC_mipi_dsi_host to the display module:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (OFF): After tearing effect request, set_tear_on is issued to display module ■ 0x1 (ON): After tearing effect request, set_tear_scan_line issued to display module <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3:2			Reserved Field: Yes

Bits	Name	Memory Access	Description
1	hw_tear_effect_gen	R/W	<p>When hw_tear_effect = 1 this bit changes tear effect by Hardware priorities:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (OFF): After tearing effect request, tear effect protocol is performed immediately after eDPI data is sent ■ 0x1 (ON): After tearing effect request, tear effect protocol is performed after eDPI data and generic commands are sent <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
0	hw_tear_effect_on	R/W	<p>This bit activates Tearing effect by hardware:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (OFF): Tearing effect request is triggered by a set_tear_on or set_tear_scanline ■ 0x1 (ON): Tearing effect request is triggered by the assertion of input pin tear_request. Note that before using tear_request the tear effect need to be active in DSI host and device. This is accomplished sending set_tear_on or set_tear_scanline commands according to DSI protocol <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.67 VID_MODE_CFG_ACT

- **Name:** VID_MODE_CFG.
- **Description:** Holds the value that controller is using for VID_MODE_CFG.
- **Size:** 32 bits
- **Offset:** 0x138
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

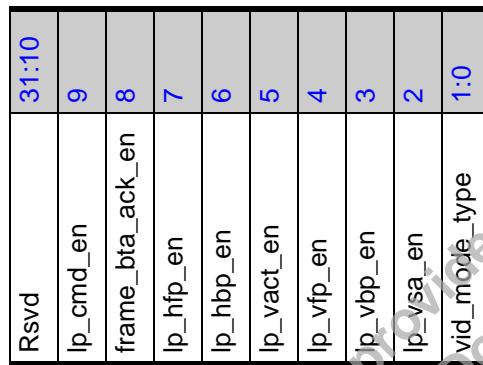


Table 5-71 Fields for Register: VID_MODE_CFG_ACT

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9	lp_cmd_en	R	When 1, this bit enables the command transmission only in low-power mode. Value After Reset: 0x0 Exists: Always
8	frame_bta_ack_en	R	When 1, this bit enables the request for an acknowledgement response at the end of a frame. Value After Reset: 0x0 Exists: Always
7	lp_hfp_en	R	When 1, this bit enables the return to low-power inside the HFP period when timing allows. Value After Reset: 0x0 Exists: Always
6	lp_hbp_en	R	When 1, this bit enables the return to low-power inside the HBP period when timing allows. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
5	lp_vact_en	R	<p>When 1, this bit enables the return to low-power inside the VACT period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	lp_vfp_en	R	<p>When 1, this bit enables the return to low-power inside the VFP period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	lp_vbp_en	R	<p>When 1, this bit enables the return to low-power inside the VBP period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	lp_vsa_en	R	<p>When 1, this bit enables the return to low-power inside the VSA period when timing allows.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1:0	vid_mode_type	R	<p>This field specifies the video mode transmission type as follows:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (VIDMODE0): Non-burst with sync pulses ■ 0x1 (VIDMODE1): Non-burst with sync events ■ 0x2 (VIDMODE2): Burst mode ■ 0x3 (VIDMODE3): Burst mode <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.68 VID_PKT_SIZE_ACT

- **Name:** Actual VID_PKT_SIZE.
- **Description:** Holds the value that controller is using for VID_PKT_SIZE.
- **Size:** 32 bits
- **Offset:** 0x13c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

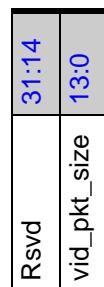


Table 5-72 Fields for Register: VID_PKT_SIZE_ACT

Bits	Name	Memory Access	Description
31:14			Reserved Field: Yes
13:0	vid_pkt_size	R	<p>This field specifies the number of pixels in a single video packet. For 18-bit not loosely packed this number must be a multiple of 4 and YCbCr Data Types, it must be a multiple of 2, as described in DSI specification.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.69 VID_NUM_CHUNKS_ACT

- **Name:** Actual VID_NUM_CHUNKS.
- **Description:** This register holds the value that controller is using for VID_NUM_CHUNKS.
- **Size:** 32 bits
- **Offset:** 0x140
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1



Table 5-73 Fields for Register: VID_NUM_CHUNKS_ACT

Bits	Name	Memory Access	Description
31:13			Reserved Field: Yes
12:0	vid_num_chunks	R	This register specifies the number of chunks to be transmitted during a Line period (a chunk is pair made of a video packet and a null packet). If set to 0 or 1, video line is still transmitted in a single packet. If set to 1 that packet is part of a chunk, meaning that a null packet follows it (if vid_null_size>0). Otherwise, multiple chunks are used to transmit each video line. Value After Reset: 0x0 Exists: Always

5.1.70 VID_NULL_SIZE_ACT

- **Name:** Actual VID_NULL_SIZE.
- **Description:** Holds the value that controller is using for VID_NULL_SIZE.
- **Size:** 32 bits
- **Offset:** 0x144
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

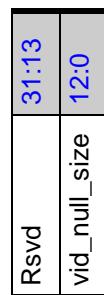


Table 5-74 Fields for Register: VID_NULL_SIZE_ACT

Bits	Name	Memory Access	Description
31:13			Reserved Field: Yes
12:0	vid_null_size	R	This register specifies the number of bytes inside a null packet. Setting to 0 disables null packets. Value After Reset: 0x0 Exists: Always

5.1.71 VID_HSA_TIME_ACT

- **Name:** Actual VID_HSA_TIME.
- **Description:** Holds the value that controller is using for VID_HSA_TIME.
- **Size:** 32 bits
- **Offset:** 0x148
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

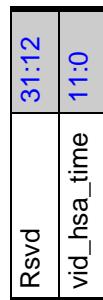


Table 5-75 Fields for Register: VID_HSA_TIME_ACT

Bits	Name	Memory Access	Description
31:12			Reserved Field: Yes
11:0	vid_hsa_time	R	This field specifies the Horizontal Synchronism Active period in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.72 VID_HBP_TIME_ACT

- **Name:** Actual VID_HBP_TIME.
- **Description:** Holds the value that controller is using for VID_HBP_TIME.
- **Size:** 32 bits
- **Offset:** 0x14c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

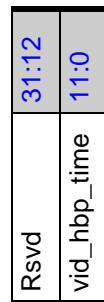


Table 5-76 Fields for Register: VID_HBP_TIME_ACT

Bits	Name	Memory Access	Description
31:12			Reserved Field: Yes
11:0	vid_hbp_time	R	This field specifies the Horizontal Back Porch period in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.73 VID_HLINE_TIME_ACT

- **Name:** Actual VID_HLINE_TIME.
- **Description:** Holds the value that controller is using for VID_HLINE_TIME.
- **Size:** 32 bits
- **Offset:** 0x150
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

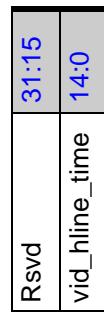


Table 5-77 Fields for Register: VID_HLINE_TIME_ACT

Bits	Name	Memory Access	Description
31:15			Reserved Field: Yes
14:0	vid_hline_time	R	This field specifies the size of the total line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles. Value After Reset: 0x0 Exists: Always

5.1.74 VID_VSA_LINES_ACT

- **Name:** Actual VID_VSA_LINES.
- **Description:** Holds the value that controller is using for VID_VSA_LINES.
- **Size:** 32 bits
- **Offset:** 0x154
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

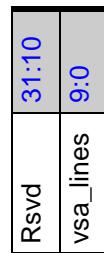


Table 5-78 Fields for Register: VID_VSA_LINES_ACT

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9:0	vsa_lines	R	This field specifies the Vertical Synchronism Active period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.75 VID_VBP_LINES_ACT

- **Name:** VID_VBP_LINES.
- **Description:** Holds the value that controller is using for VID_VBP_LINES.
- **Size:** 32 bits
- **Offset:** 0x158
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

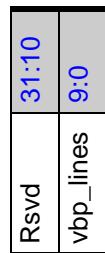


Table 5-79 Fields for Register: VID_VBP_LINES_ACT

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9:0	vbp_lines	R	This field specifies the Vertical Back Porch period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.76 VID_VFP_LINES_ACT

- **Name:** Actual VID_VFP_LINES.
- **Description:** Holds the value that controller is using for VID_VFP_LINES.
- **Size:** 32 bits
- **Offset:** 0x15c
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

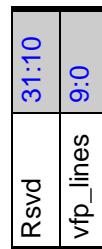


Table 5-80 Fields for Register: VID_VFP_LINES_ACT

Bits	Name	Memory Access	Description
31:10			Reserved Field: Yes
9:0	vfp_lines	R	This field specifies the Vertical Front Porch period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.77 VID_VACTIVE_LINES_ACT

- **Name:** Actual VID_VACTIVE_LINES.
- **Description:** Holds the value that controller is using for VID_VACTIVE_LINES.
- **Size:** 32 bits
- **Offset:** 0x160
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

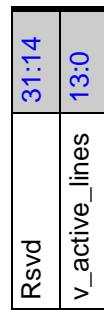


Table 5-81 Fields for Register: VID_VACTIVE_LINES_ACT

Bits	Name	Memory Access	Description
31:14			Reserved Field: Yes
13:0	v_active_lines	R	This field specifies the Vertical Active period measured in number of horizontal lines. Value After Reset: 0x0 Exists: Always

5.1.78 VID PKT STATUS

- **Name:** eDPI and DPI FIFOs status.
 - **Description:** Contains information about the status of FIFOs related to DPI and eDPI interfaces.
 - **Size:** 32 bits
 - **Offset:** 0x168
 - **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

Rsvd	31:24
edpi_buff_pld_full	23
edpi_buff_pld_empty	22
edpi_buff_cmd_full	21
edpi_buff_cmd_empty	20
Rsvd	19:18
dpi_buff_pld_full	17
dpi_buff_pld_empty	16
Rsvd	15:8
edpi_pld_w_full	7
edpi_pld_w_empty	6
edpi_Cmd_w_full	5
edpi_Cmd_w_empty	4
dpi_pld_w_full	3
dpi_pld_w_empty	2
dpi_cmd_w_full	1
dpi_cmd_w_empty	0

Table 5-82 Fields for Register: VID_PKT_STATUS

Bits	Name	Memory Access	Description
31:24			Reserved Field: Yes
23	edpi_buff_pld_full	R	<p>This bit indicates the full status of the edpi payload internal buffer. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
22	edpi_buff_pld_empty	R	<p>This bit indicates the empty status of the edpi payload internal buffer. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x1</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
21	edpi_buff_cmd_full	R	<p>This bit indicates the full status of the edpi command internal buffer. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
20	edpi_buff_cmd_empty	R	<p>This bit indicates the empty status of the edpi command internal buffer. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x1</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
19:18			Reserved Field: Yes

Bits	Name	Memory Access	Description
17	dpi_buff_pld_full	R	<p>This bit indicates the full status of the payload internal buffer for video Mode. This bit is set to 0 for command Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
16	dpi_buff_pld_empty	R	<p>This bit indicates the empty status of the payload internal buffer for video Mode. This bit is set to 0 for command Mode.</p> <p>Value After Reset: "(DSI_HOST_EDPIINTERFACE== 1) ? 0 : 1"</p> <p>Exists: Always</p>
15:8			Reserved Field: Yes
7	edpi_pld_w_full	R	<p>This bit indicates the full status of write payload FIFO for command Mode. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
6	edpi_pld_w_empty	R	<p>This bit indicates the empty status of write payload FIFO for command Mode. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x1</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
5	edpi_cmd_w_full	R	<p>This bit indicates the full status of write command FIFO for command Mode. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
4	edpi_cmd_w_empty	R	<p>This bit indicates the empty status of write command FIFO for command Mode. This bit is set to 0 for Video Mode.</p> <p>Value After Reset: 0x1</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
3	dpi_pld_w_full	R	<p>This bit indicates the full status of write payload FIFO for video Mode. This bit is set to 0 for command Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	dpi_pld_w_empty	R	<p>This bit indicates the empty status of write payload FIFO for video Mode. This bit is set to 0 for command Mode.</p> <p>Value After Reset: "(DSI_HOST_EDPIINTERFACE== 1) ? 0 : 1"</p> <p>Exists: Always</p>
1	dpi_cmd_w_full	R	<p>This bit indicates the full status of write command FIFO for video Mode. This bit is set to 0 for command Mode.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
0	dpi_cmd_w_empty	R	<p>This bit indicates the empty status of write command FIFO for video Mode. This bit is set to 0 for command Mode.</p> <p>Value After Reset: "(DSI_HOST_EDPIINTERFACE== 1) ? 0 : 1"</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.79 SDF_3D_ACT

- **Name:** SDF_3D.
- **Description:** Holds the value that controller is using for SDF_3D.
- **Size:** 32 bits
- **Offset:** 0x190
- **Exists:** ((DSI_HOST_DATAINTERFACE == 2) || (DSI_HOST_DATAINTERFACE == 3) || (DSI_HOST_DATAINTERFACE == 4)) == 1

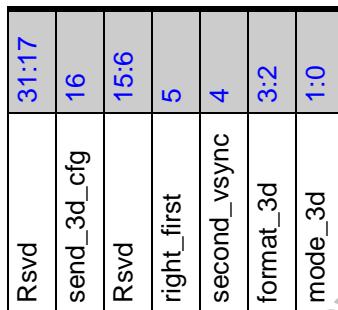


Table 5-83 Fields for Register: SDF_3D_ACT

Bits	Name	Memory Access	Description
31:17			Reserved Field: Yes
16	send_3d_cfg	R	When set, causes the next VSS packet to include 3D control payload in every VSS packet. Value After Reset: 0x0 Exists: Always
15:6			Reserved Field: Yes
5	right_first	R	This bit specifies the left/right order: Values: <ul style="list-style-type: none"> ■ 0x0 (LEFTFIRST): left eye is sent first, then right eye ■ 0x1 (RIGHTFIRST): right eye data is sent first, then left eye Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
4	second_vsync	R	<p>This field specifies whether there is a second VSYNC pulse between Left and Right Images, when 3D Image Format is Frame-based:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (NOSYNCPULSE): No sync pulses between left and right data. ■ 0x1 (SYNCPULSE): Sync pulse (HSYNC, VSYNC, blanking) between left and right data. <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3:2	format_3d	R	<p>This field specifies 3D Image Format:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (LINE): Alternating lines of left and right data ■ 0x1 (FRAME): Alternating frames of left and right data ■ 0x2 (PIXEL): Alternating pixels of left and right data ■ 0x3 (RESERVED): Reserved, not used <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1:0	mode_3d	R	<p>This field specifies 3D Mode On/Off and Display Orientation:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (MODE0): 3D Mode Off, 2D Mode On ■ 0x1 (MODE1): 3D Mode On, Portrait Orientation ■ 0x2 (MODE2): 3D Mode On, Landscape Orientation ■ 0x3 (MODE3): Reserved, not used <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.80 DSC_ENC_COREID

- **Name:** COREID.
- **Description:** Indicates DSC encoder COREID.
- **Size:** 32 bits
- **Offset:** 0x200
- **Exists:** DSI_HOST_DSC_ENC==1

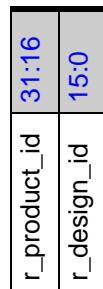


Table 5-84 Fields for Register: DSC_ENC_COREID

Bits	Name	Memory Access	Description
31:16	r_product_id	R	VESA DSC Encoder version (ie: v1.1 value is 1100). Value After Reset: 0x1100 Exists: Always
15:0	r_design_id	R	Design version. (ie: v1.00a value is 0x100A). Value After Reset: 0x100a Exists: Always

5.1.81 DSC_ENC_VERSION

- **Name:** DSC Version.
- **Description:** Indicates Vesa DSC Version.
- **Size:** 32 bits
- **Offset:** 0x204
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-85 Fields for Register: DSC_ENC_VERSION

Bits	Name	Memory Access	Description
31:0	r_dsc_version	R	Vesa DSC Version. Value After Reset: 0x3130302a Exists: Always

5.1.82 DSC_ENC_FLATNESS_DET_THRES

- **Name:** DSC encoder flatness.
- **Description:** DSC encoder flatness determination.
- **Size:** 32 bits
- **Offset:** 0x208
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-86 Fields for Register: DSC_ENC_FLATNESS_DET_THRES

Bits	Name	Memory Access	Description
31:13			Reserved Field: Yes
12:0	flatness_det_thres	R/W	Threshold between "somewhat flat" and "very flat" for flatness determination. Value After Reset: 0x2 Exists: Always

5.1.83 DSC_ENC_DELAY

- **Name:** DPI events delay.
- **Description:** Used to delay DPI events in order to be compliant with DSC encoder latency.
- **Size:** 32 bits
- **Offset:** 0x20c
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-87 Fields for Register: DSC_ENC_DELAY

Bits	Name	Memory Access	Description
31:0	dsc_enc_delay	R/W	<p>This register is use to delay DPI events in order to be compliant with DSC encoder latency.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.84 DSC_ENC_COMPRESSED_LINE_SIZE

- **Name:** Compressed line size.
- **Description:** Compressed line size in units of 1/16th of a bit.
- **Size:** 32 bits
- **Offset:** 0x210
- **Exists:** DSI_HOST_DSC_ENC==1

	31:26	
Rsvd		25:0
	compressed_line_size_frac_bits	

Table 5-88 Fields for Register: DSC_ENC_COMPRESSED_LINE_SIZE

Bits	Name	Memory Access	Description
31:26			Reserved Field: Yes
25:0	compressed_line_size_frac_bits	R/W	<p>Contains the size of a compressed line, in units of 1/16th of a bit. Value is programmed to rb_bits_per_pixel_r * slice_width, where rb_bits_per_pixel_r is an integer expressed in units of 1/16th of a bit per pixel.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.85 DSC_ENC_LINES_IN_EXCESS

- **Name:** Number of lines in excess.
- **Description:** Calculation of the number of lines in excess.
- **Size:** 32 bits
- **Offset:** 0x214
- **Exists:** DSI_HOST_DSC_ENC==1

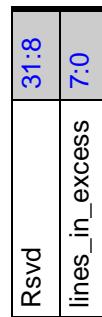


Table 5-89 Fields for Register: DSC_ENC_LINES_IN_EXCESS

Bits	Name	Memory Access	Description
31:8			Reserved Field: Yes
7:0	lines_in_excess	R/W	Number of additional lines that need to be output, due to extra bits sent in dsc_data[63:0] at the end of each line. Value After Reset: 0x0 Exists: Always

5.1.86 DSC_ENC_RBUF_ADDR_LAST_LINE_ADJ

- **Name:** DSC adjustment.
- **Description:** Adjustment needed to calculate end of last line of a slice.
- **Size:** 32 bits
- **Offset:** 0x218
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-90 Fields for Register: DSC_ENC_RBUF_ADDR_LAST_LINE_ADJ

Bits	Name	Memory Access	Description
31:27			Reserved Field: Yes
26:0	rbuf_addr_last_line_adj	R/W	<p>Value of adjustment, determined iteratively, that needs to be applied to the calculation of the end address of last line of a slice in the RAM of rate buffer.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.87 DSC_MODE

- **Name:** DSC encoder.
- **Description:** Enable DSC encoder.
- **Size:** 32 bits
- **Offset:** 0x21c
- **Exists:** DSI_HOST_DSC_ENC==1

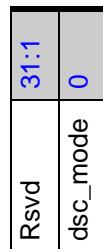


Table 5-91 Fields for Register: DSC_MODE

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	dsc_mode	R/W	<p>This register configures DSC mode:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (DSC): DSC 24 ■ 0x1 (DSCENCODER): DSC encoder <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.88 DSC_ENC_INT_ST

- **Name:** DSC encoder interrupt status.
- **Description:** Contains the status of interrupt sources from DSC encoder.
- **Size:** 32 bits
- **Offset:** 0x220
- **Exists:** DSI_HOST_DSC_ENC==1

Rsvd	31:20	19	demux_underflow1
		18	demux_overflow1
		17	demux_underflow0
		16	demux_overflow0
		15	dpi_mpb_delay_irq
		14	lb_ram_underflow_err
		13	lb_ram_overflow_err
		12	bal_fifo_cg_underflow_err
		11	bal_fifo_cg_overflow_err
		10	bal_fifo_co_underflow_err
		9	bal_fifo_co_overflow_err
		8	bal_fifo_y_underflow_err
		7	bal_fifo_y_overflow_err
		6	se_fifo_underflow_err
		5	se_fifo_overflow_err
		4	rate_buf_underflow_err
		3	rate_buf_overflow_err
		2	slice_length_err
		1	bm_underflow_err
		0	rc_overflow_err

Table 5-92 Fields for Register: DSC_ENC_INT_ST

Bits	Name	Memory Access	Description
31:20			Reserved Field: Yes
19	demux_underflow1	R	Underflow in the demux RAM for DSC Encoder 1. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
18	demux_overflow1	R	Overflow in the demux RAM for DSC Encoder 1. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
17	demux_underflow0	R	Underflow in the demux RAM for DSC Encoder 0. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
16	demux_overflow0	R	Overflow in the demux RAM for DSC Encoder 0. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
15	dpi_mpb_delay_irq	R	Overflow in the event delay buffer. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
14	lb_ram_underflow_err	R	<p>Line Buffer RAM underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	lb_ram_overflow_err	R	<p>Line Buffer RAM overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
12	bal_fifo_cg_underflow_err	R	<p>Balance FIFO for channel Cg underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	bal_fifo_cg_overflow_err	R	<p>Balance FIFO for channel Cg overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	bal_fifo_co_underflow_err	R	<p>Balance FIFO for channel Co underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	bal_fifo_co_overflow_err	R	<p>Balance FIFO for channel Co overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	bal_fifo_y_underflow_err	R	<p>Balance FIFO for channel Y underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	bal_fifo_y_overflow_err	R	<p>Balance FIFO for channel Y overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
6	se_fifo_underflow_err	R	<p>Syntax Element FIFO underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5	se_fifo_overflow_err	R	<p>Syntax Element FIFO overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	rate_buf_underflow_err	R	<p>Rate Buffer FIFO underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	rate_buf_overflow_err	R	<p>Rate Buffer FIFO overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
2	slice_length_err	R	Slice length does not match PPS settings. Value After Reset: 0x0 Exists: Always
1	bm_underflow_err	R	Buffer model underflow. Value After Reset: 0x0 Exists: Always
0	rc_overflow_err	R	RC Buffer Fullness overflow. Value After Reset: 0x0 Exists: Always

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.89 DSC ENC INT MSK

- **Name:** INT_ST_DSC mask.
 - **Description:** Configures masks for the sources of interrupts that affect the INT_ST_DSC register. Write 1 to un-mask each error report.
 - **Size:** 32 bits
 - **Offset:** 0x224
 - **Exists:** DSI HOST DSC ENC==1

Rsvd	31:20
demux_underflow1_msk	19
demux_overflow1_msk	18
demux_underflow0_msk	17
demux_overflow0_msk	16
dpi_mpbi_delay_irq_msk	15
lb_ram_underflow_err_msk	14
lb_ram_overflow_err_msk	13
bal_fifo_cg_underflow_err_msk	12
bal_fifo_cg_overflow_err_msk	11
bal_fifo_co_underflow_err_msk	10
bal_fifo_co_overflow_err_msk	9
bal_fifo_y_underflow_err_msk	8
bal_fifo_y_overflow_err_msk	7
se_fifo_underflow_err_msk	6
se_fifo_overflow_err_msk	5
rate_buf_underflow_err_msk	4
rate_buf_overflow_err_msk	3
slice_length_err_msk	2
bm_underflow_err_msk	1
rc_overflow_err_msk	0

Table 5-93 Fields for Register: DSC_ENC_INT_MSK

Bits	Name	Memory Access	Description
31:20			Reserved Field: Yes
19	demux_underflow1_msk	R/W	<p>Write 1 to un-mask the underflow in the demux RAM for DSC Encoder 1 error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1</p>
18	demux_overflow1_msk	R/W	<p>Write 1 to un-mask the overflow in the demux RAM for DSC Encoder 1 error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1</p>
17	demux_underflow0_msk	R/W	<p>Write 1 to un-mask the underflow in the demux RAM for DSC Encoder 0 error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1</p>

Bits	Name	Memory Access	Description
16	demux_overflow0_msk	R/W	<p>Write 1 to un-mask the overflow in the demux RAM for DSC Encoder 0 error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1</p>
15	dpi_mpmb_delay_irq_msk	R/W	<p>Write 1 to un-mask the overflow in the event delay buffer error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
14	lb_ram_underflow_err_msk	R/W	<p>Write 1 to un-mask the Line Buffer RAM underflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	lb_ram_overflow_err_msk	R/W	<p>Write 1 to un-mask the Line Buffer RAM overflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
12	bal_fifo_cg_underflow_err_msk	R/W	<p>Write 1 to un-mask the Balance FIFO for channel Cg underflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	bal_fifo_cg_overflow_err_msk	R/W	<p>Write 1 to un-mask the Balance FIFO for channel Cg overflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	bal_fifo_co_underflow_err_msk	R/W	<p>Write 1 to un-mask the Balance FIFO for channel Co underflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	bal_fifo_co_overflow_err_msk	R/W	<p>Write 1 to un-mask the Balance FIFO for channel Co overflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	bal_fifo_y_underflow_err_msk	R/W	<p>Write 1 to un-mask the Balance FIFO for channel Y underflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	bal_fifo_y_overflow_err_msk	R/W	<p>Write 1 to un-mask the Balance FIFO for channel Y overflow report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
6	se_fifo_underflow_err_msk	R/W	Write 1 to un-mask the Syntax Element FIFO underflow report. Value After Reset: 0x0 Exists: Always
5	se_fifo_overflow_err_msk	R/W	Write 1 to un-mask the Syntax Element FIFO overflow report. Value After Reset: 0x0 Exists: Always
4	rate_buf_underflow_err_msk	R/W	Write 1 to un-mask the Rate Buffer FIFO underflow report. Value After Reset: 0x0 Exists: Always
3	rate_buf_overflow_err_msk	R/W	Write 1 to un-mask the Rate Buffer FIFO overflow report. Value After Reset: 0x0 Exists: Always
2	slice_length_err_msk	R/W	Write 1 to un-mask the Slice length does not match PPS settings report. Value After Reset: 0x0 Exists: Always
1	bm_underflow_err_msk	R/W	Write 1 to un-mask the Buffer model underflow report. Value After Reset: 0x0 Exists: Always
0	rc_overflow_err_msk	R/W	Write 1 to un-mask the RC Buffer Fullness overflow report. Value After Reset: 0x0 Exists: Always

5.1.90 DSC_ENC_INT_FORCE

- **Name:** INT_ST_DSC interrupt force.
- **Description:** Forces interrupts that affect the INT_ST_DSC register.
- **Size:** 32 bits
- **Offset:** 0x228
- **Exists:** DSI_HOST_DSC_ENC==1

Rsvd	31:20
demux_underflow1_force	19
demux_overflow1_force	18
demux_underflow0_force	17
demux_overflow0_force	16
dpi_mpib_delay_irq_force	15
lb_ram_underflow_err_force	14
lb_ram_overflow_err_force	13
bal_fifo_cg_underflow_err_force	12
bal_fifo_cg_overflow_err_force	11
bal_fifo_co_underflow_err_force	10
bal_fifo_co_overflow_err_force	9
bal_fifo_y_underflow_err_force	8
bal_fifo_y_overflow_err_force	7
se_fifo_underflow_err_force	6
se_fifo_overflow_err_force	5
rate_buf_underflow_err_force	4
rate_buf_overflow_err_force	3
slice_length_err_force	2
bm_underflow_err_force	1
rc_overflow_err_force	0

Table 5-94 Fields for Register: DSC_ENC_INT_FORCE

Bits	Name	Memory Access	Description
31:20			Reserved Field: Yes
19	demux_underflow1_force	R/WC	This register forces interrupt of underflow in the demux RAM for DSC Encoder 1. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
18	demux_overflow1_force	R/WC	This register forces interrupt of overflow in the demux RAM for DSC Encoder 1. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
17	demux_underflow0_force	R/WC	This register forces interrupt of underflow in the demux RAM for DSC Encoder 0. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1

Bits	Name	Memory Access	Description
16	demux_overflow0_force	R/WC	<p>This register forces interrupt of overflow in the demux RAM for DSC Encoder 0.</p> <p>Value After Reset: 0x0</p> <p>Exists: DS1_HOST_DUAL_DSC_ENC_INCDEMUX==1</p>
15	dpi_mpb_delay_irq_force	R/WC	<p>This register forces interrupt of overflow in the event delay buffer.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
14	lb_ram_underflow_err_force	R/WC	<p>Force Line Buffer RAM underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	lb_ram_overflow_err_force	R/WC	<p>Force Line Buffer RAM overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
12	bal_fifo_cg_underflow_err_force	R/WC	<p>Force Balance FIFO for channel Cg underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	bal_fifo_cg_overflow_err_force	R/WC	<p>Force Balance FIFO for channel Cg overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	bal_fifo_co_underflow_err_force	R/WC	<p>Force Balance FIFO for channel Co underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	bal_fifo_co_overflow_err_force	R/WC	<p>Force Balance FIFO for channel Co overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	bal_fifo_y_underflow_err_force	R/WC	<p>Force Balance FIFO for channel Y underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	bal_fifo_y_overflow_err_force	R/WC	<p>Force Balance FIFO for channel Y overflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
6	se_fifo_underflow_err_force	R/WC	<p>Force Syntax Element FIFO underflow.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
5	se_fifo_overflow_err_force	R/WC	Force Syntax Element FIFO overflow. Value After Reset: 0x0 Exists: Always
4	rate_buf_underflow_err_force	R/WC	Force Rate Buffer FIFO underflow. Value After Reset: 0x0 Exists: Always
3	rate_buf_overflow_err_force	R/WC	Force Rate Buffer FIFO overflow. Value After Reset: 0x0 Exists: Always
2	slice_length_err_force	R/WC	Force Slice length does not match PPS settings error. Value After Reset: 0x0 Exists: Always
1	bm_underflow_err_force	R/WC	Force Buffer model underflow. Value After Reset: 0x0 Exists: Always
0	rc_overflow_err_force	R/WC	Force RC Buffer Fullness overflow. Value After Reset: 0x0 Exists: Always

5.1.91 DSC_FIFO_STATUS_SELECT

- **Name:** DSC FIFO selection.
- **Description:** Selection of the DSC FIFO whose word count is reported in DSC_FIFO_STATUS register.
- **Size:** 32 bits
- **Offset:** 0x22c
- **Exists:** DSI_HOST_DSC_ENC==1

Rsvd	31:3	2:0
		dsc_fifo_status_select

Table 5-95 Fields for Register: DSC_FIFO_STATUS_SELECT

Bits	Name	Memory Access	Description
31:3			Reserved Field: Yes
2:0	dsc_fifo_status_select	R/W	<p>Selection of the DSC FIFO whose word count is reported in DSC_FIFO_STATUS register.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (RATEBUFF): Rate Buffer word count. ■ 0x1 (SELEMENT): Syntax Element word count. ■ 0x2 (BFIFO0): Balance FIFO 0 word count. ■ 0x3 (BFIFO1): Balance FIFO 1 word count. ■ 0x4 (BFIF2): Balance FIFO 2 word count. ■ 0x5 (LBUFFER): Line Buffer word count. ■ 0x6 (SDMUX): Slice Demultiplexer FIFO word count. If DSI_HOST_DUAL_DSC_ENC_INCDEMUX == 0, then this value is read as zero. <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.92 DSC_FIFO_STATUS

- **Name:** DSC encoder FIFOs status.
- **Description:** Contains information on the status of FIFOs related to DSC encoder.
- **Size:** 32 bits
- **Offset:** 0x230
- **Exists:** DSI_HOST_DSC_ENC==1

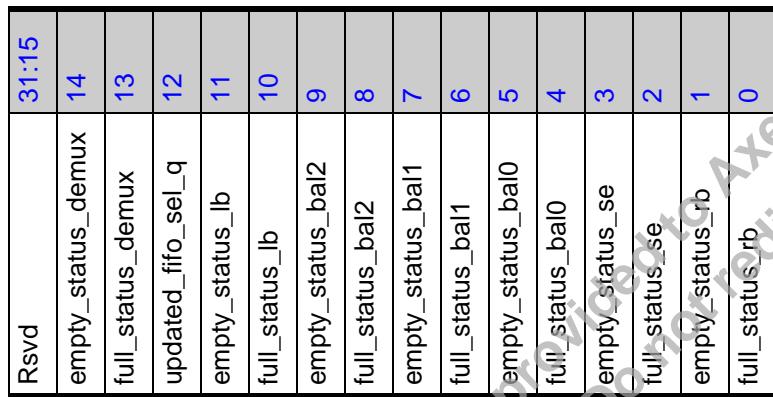


Table 5-96 Fields for Register: DSC_FIFO_STATUS

Bits	Name	Memory Access	Description
31:15			Reserved Field: Yes
14	empty_status_demux	R	Empty status bit for Slice Demultiplexer FIFO. Value After Reset: 0x1 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
13	full_status_demux	R	Full status bit for Slice Demultiplexer FIFO. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
12	updated_fifo_sel_q	R	Flag indicating whether the reported word count effectively corresponds to the FIFO currently selected in FIFO_STATUS_SELECT register. (Part of a mechanism to avoid using wrong values due to CDC issues.) Value After Reset: 0x1 Exists: Always
11	empty_status_lb	R	Empty status bit for Line buffer. Value After Reset: 0x1 Exists: Always
10	full_status_lb	R	Full status bit for Line buffer. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
9	empty_status_bal2	R	Empty status bit for Balance FIFO 2, Cg/Cr channel. Value After Reset: 0x1 Exists: Always
8	full_status_bal2	R	Full status bit for Balance FIFO 2, Cg/Cr channel. Value After Reset: 0x0 Exists: Always
7	empty_status_bal1	R	Empty status bit for Balance FIFO 1, Co/Cb channel. Value After Reset: 0x1 Exists: Always
6	full_status_bal1	R	Full status bit for Balance FIFO 1, Co/Cb channel. Value After Reset: 0x0 Exists: Always
5	empty_status_bal0	R	Empty status bit for Balance FIFO 0, Y channel. Value After Reset: 0x1 Exists: Always
4	full_status_bal0	R	Full status bit for Balance FIFO 0, Y channel. Value After Reset: 0x0 Exists: Always
3	empty_status_se	R	Empty status bit for Syntax element FIFO. Value After Reset: 0x1 Exists: Always
2	full_status_se	R	Full status bit for Syntax element FIFO. Value After Reset: 0x0 Exists: Always
1	empty_status_rb	R	Empty status bit for Rate Buffer. Value After Reset: 0x1 Exists: Always
0	full_status_rb	R	Full status bit for Rate Buffer. Value After Reset: 0x0 Exists: Always

5.1.93 DSC_FIFO_STATUS2

- **Name:** Second DSC encoder FIFOs status.
- **Description:** Contains information on the status of FIFOs related to second DSC encoder.
- **Size:** 32 bits
- **Offset:** 0x234
- **Exists:** DSI_HOST_DSC_ENC_TYPE!=1

Rsvd	31:15	empty_status_demux	14	full_status_demux	13	updated_fifo_sel_q	12	empty_status_lb	11	full_status_lb	10	empty_status_bal2	9	full_status_bal2	8	empty_status_bal1	7	full_status_bal1	6	empty_status_bal0	5	full_status_bal0	4	empty_status_se	3	full_status_se	2	empty_status_rb	1	full_status_rb	0
------	-------	--------------------	----	-------------------	----	--------------------	----	-----------------	----	----------------	----	-------------------	---	------------------	---	-------------------	---	------------------	---	-------------------	---	------------------	---	-----------------	---	----------------	---	-----------------	---	----------------	---

Table 5-97 Fields for Register: DSC_FIFO_STATUS2

Bits	Name	Memory Access	Description
31:15			Reserved Field: Yes
14	empty_status_demux	R	Empty status bit for Slice Demultiplexer FIFO. Value After Reset: 0x1 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
13	full_status_demux	R	Full status bit for Slice Demultiplexer FIFO. Value After Reset: 0x0 Exists: DSI_HOST_DUAL_DSC_ENC_INCDEMUX==1
12	updated_fifo_sel_q	R	Flag indicating whether the reported word count effectively corresponds to the FIFO currently selected in FIFO_STATUS_SELECT register. (Part of a mechanism to avoid using wrong values due to CDC issues.) Value After Reset: 0x1 Exists: Always
11	empty_status_lb	R	Empty status bit for Line buffer. Value After Reset: 0x1 Exists: Always
10	full_status_lb	R	Full status bit for Line buffer. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
9	empty_status_bal2	R	Empty status bit for Balance FIFO 2, Cg/Cr channel. Value After Reset: 0x1 Exists: Always
8	full_status_bal2	R	Full status bit for Balance FIFO 2, Cg/Cr channel. Value After Reset: 0x0 Exists: Always
7	empty_status_bal1	R	Empty status bit for Balance FIFO 1, Co/Cb channel. Value After Reset: 0x1 Exists: Always
6	full_status_bal1	R	Full status bit for Balance FIFO 1, Co/Cb channel. Value After Reset: 0x0 Exists: Always
5	empty_status_bal0	R	Empty status bit for Balance FIFO 0, Y channel. Value After Reset: 0x1 Exists: Always
4	full_status_bal0	R	Full status bit for Balance FIFO 0, Y channel. Value After Reset: 0x0 Exists: Always
3	empty_status_se	R	Empty status bit for Syntax element FIFO. Value After Reset: 0x1 Exists: Always
2	full_status_se	R	Full status bit for Syntax element FIFO. Value After Reset: 0x0 Exists: Always
1	empty_status_rb	R	Empty status bit for Rate Buffer. Value After Reset: 0x1 Exists: Always
0	full_status_rb	R	Full status bit for Rate Buffer. Value After Reset: 0x0 Exists: Always

5.1.94 DSC_FIFO_WORD_COUNT

- **Name:** DSC encoder FIFOs word count.
- **Description:** Contains information on the word count of FIFOs related to the DSC encoder.
- **Size:** 32 bits
- **Offset:** 0x238
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-98 Fields for Register: DSC_FIFO_WORD_COUNT

Bits	Name	Memory Access	Description
31:0	fifo_word_count	R	Word count of the FIFO selected using DSC_FIFO_STATUS_SELECT register. Value After Reset: 0x0 Exists: Always

5.1.95 DSC_FIFO_WORD_COUNT2

- **Name:** Second DSC encoder FIFOs word count.
- **Description:** Contains information on the word count of FIFOs related to the second DSC encoder.
- **Size:** 32 bits
- **Offset:** 0x23c
- **Exists:** DSI_HOST_DSC_ENC_TYPE!=1



Table 5-99 Fields for Register: DSC_FIFO_WORD_COUNT2

Bits	Name	Memory Access	Description
31:0	fifo_word_count	R	Word count of the FIFO selected using DSC_FIFO_STATUS_SELECT register. Value After Reset: 0x0 Exists: Always

5.1.96 DSC_ENC_PPS_0_3

- **Name:** Picture Parameter Set[0,3].
- **Description:** Holds the bytes in the range [0,3] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x260
- **Exists:** DSI_HOST_DSC_ENC==1

rb_dsc_version_major_r	31:28
rb_dsc_version_minor_r	27:24
rb_pps_identifier_r	23:16
Rsvd	15:8
rb_bits_per_component_r	7:4
rb_linebuf_depth_r	3:0

Table 5-100 Fields for Register: DSC_ENC_PPS_0_3

Bits	Name	Memory Access	Description
31:28	rb_dsc_version_major_r	R/W	Contains the major version of DSC. Value After Reset: 0x0 Exists: Always
27:24	rb_dsc_version_minor_r	R/W	Contains the minor version of DSC. Value After Reset: 0x0 Exists: Always
23:16	rb_pps_identifier_r	R/W	Application-specific identifier that can be used to differentiate between different PPS tables. If PPS transmission is not defined by an application specification, the value should be 0x00 . Value After Reset: 0x0 Exists: Always
15:8			Reserved Field: Yes

Bits	Name	Memory Access	Description
7:4	rb_bits_per_component_r	R/W	<p>Indicates the number of bits per component for the original pixels of the encoded picture (All other encodings are reserved):</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x8 (BPC0): 8bpc ■ 0xa (BCP1): 10bpc ■ 0xc (BCP2): 12bpc <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3:0	rb_linebuf_depth_r	R/W	<p>Contains the line buffer bit depth used to generate the bitstream. If a component's bit depth (see section 6.1 of VESA DSC Standard v1.1) is greater than this value, the line storage rounds the reconstructed values to this number of bits (All other encodings are reserved):</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x8 (ENC0): 8 bits ■ 0x9 (ENC1): 9 bits ■ 0xa (ENC2): 10 bits ■ 0xb (ENC3): 11 bits ■ 0xc (ENC4): 12 bits ■ 0xd (ENC5): 13 bits <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.97 DSC_ENC_PPS_4_7

- **Name:** Picture Parameter Set[4,7].
- **Description:** Holds the bytes in the range [4,7] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x264
- **Exists:** DSI_HOST_DSC_ENC==1

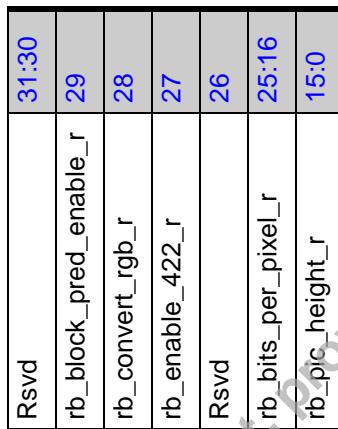


Table 5-101 Fields for Register: DSC_ENC_PPS_4_7

Bits	Name	Memory Access	Description
31:30			Reserved Field: Yes
29	rb_block_pred_enable_r	R/W	<p>Selects whether Block Prediction is used:- 1: ACTIVE (Decoder must select between BP and MMAP, using the method described in section "Selection between Block and Modified Median-Adaptive Prediction" of VESA DSC specification.</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (INACTIVE): BP is not used to code any groups within the picture <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
28	rb_convert_rgb_r	R/W	<p>Indicates whether DSC color space conversion is active:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (INACTIVE): Color space is YCbCr. ■ 0x1 (ACTIVE): Encoder converts RGB to YCoCg-R, and decoder converts YCoCg-R to RGB. <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
27	rb_enable_422_r	R/W	<p>Indicates the chroma sampling of the input data:</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (SAMP0): Input uses 4:4:4 sampling ■ 0x1 (SAMP1): Input uses 4:2:2 sampling <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
26			Reserved Field: Yes
25:16	rb_bits_per_pixel_r	R/W	<p>Specifies the target bits/pixel (bpp) rate that is used by the encoder, in steps of 1/16 of a bit per pixel. Only values greater than or equal to 6bpp are allowed. If vbr_enable is cleared to 0, this value must be less than or equal to the sustained rate that would apply if MPP is always selected, which is a function of bits_per_component, convert_rgb, and rc_range_parameters[0].</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:0	rb_pic_height_r	R/W	<p>Specify the picture size, in units of pixels. pic_height is the number of pixel rows within the raster. Although not required, it is recommended that pic_height and pic_width be close to integer multiples of slice_height and slice_width, respectively.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.98 DSC_ENC_PPS_8_11

- **Name:** Picture Parameter Set[8,11].
- **Description:** Holds the bytes in the range [8,11] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x268
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-102 Fields for Register: DSC_ENC_PPS_8_11

Bits	Name	Memory Access	Description
31:16	rb_pic_width_r	R/W	<p>Specify the picture size, in units of pixels. pic_width is the number of pixel columns within the raster. Although not required, it is recommended that pic_height and pic_width be close to integer multiples of slice_height and slice_width, respectively.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:0	rb_slice_height_r	R/W	<p>Specify the size for each slice, in units of pixels. All slices that comprise a single picture are required to have an identical size. If the pic_height is not evenly divisible by the slice_height, lines consisting of midpoint-valued samples are added to the bottommost slices so that these slices are the same height as the other slices. The transport must allocate transmission time for sending the compressed bits corresponding to any replicated pixels.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.99 DSC_ENC_PPS_12_15

- **Name:** Picture Parameter Set[12,15].
- **Description:** Holds the bytes in the range [12,15] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x26c
- **Exists:** DSI_HOST_DSC_ENC==1

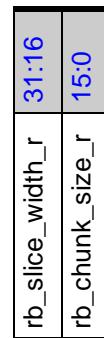


Table 5-103 Fields for Register: DSC_ENC_PPS_12_15

Bits	Name	Memory Access	Description
31:16	rb_slice_width_r	R/W	<p>Specify the size for each slice, in units of pixels. All slices that comprise a single picture are required to have an identical size. If the pic_width is not evenly divisible by the slice_width, the rightmost column of pixels is replicated to pad the rightmost slices to be the same width as the other slices. The transport must allocate transmission time for sending the compressed bits corresponding to any replicated pixels.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:0	rb_chunk_size_r	R/W	<p>Indicates the size, in units of bytes, of the chunks that are used for slice multiplexing (see section 4.2.2 of VESA DSC Standard v1.1). If vbr_enable is set to 1, this is the maximum size of the chunks. Value is programmed to ceil(bits_per_pixel * slice_width / 8) bytes.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.100 DSC_ENC_PPS_16_19

- **Name:** Picture Parameter Set[16,19].
- **Description:** Holds the bytes in the range [16,19] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x270
- **Exists:** DSI_HOST_DSC_ENC==1

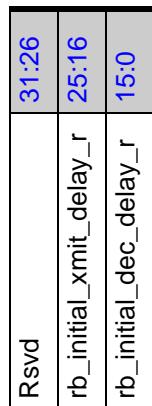


Table 5-104 Fields for Register: DSC_ENC_PPS_16_19

Bits	Name	Memory Access	Description
31:26			Reserved Field: Yes
25:16	rb_initial_xmit_delay_r	R/W	Initial transmission delay. Specifies the number of pixel times that the encoder waits before transmitting data from its rate buffer. Value After Reset: 0x0 Exists: Always
15:0	rb_initial_dec_delay_r	R/W	Initial decoding delay. Specifies the number of pixel times that the decoder accumulates data in its rate buffer before starting to decode and output pixels. Value After Reset: 0x0 Exists: Always

5.1.101 DSC_ENC_PPS_20_23

- **Name:** Picture Parameter Set[20,23].
- **Description:** Holds the bytes in the range [20,23] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x274
- **Exists:** DSI_HOST_DSC_ENC==1

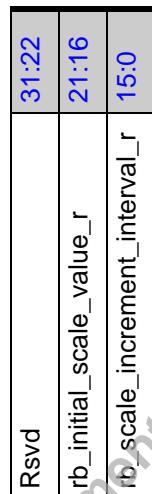


Table 5-105 Fields for Register: DSC_ENC_PPS_20_23

Bits	Name	Memory Access	Description
31:22			Reserved Field: Yes
21:16	rb_initial_scale_value_r	R/W	Specifies the initial rcXformScale factor value used at the beginning of a slice (see section 6.8.2 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
15:0	rb_scale_increment_interval_r	R/W	Specifies the number of group times between incrementing the rcXformScale factor at the end of a slice (see section 6.8.2 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

5.1.102 DSC_ENC_PPS_24_27

- **Name:** Picture Parameter Set[24,27].
- **Description:** Holds the bytes in the range [24,27] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x278
- **Exists:** DSI_HOST_DSC_ENC==1

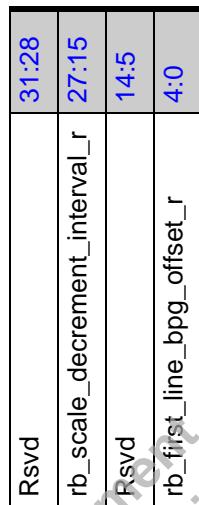


Table 5-106 Fields for Register: DSC_ENC_PPS_24_27

Bits	Name	Memory Access	Description
31:28			Reserved Field: Yes
27:15	rb_scale_decrement_interval_r	R/W	Specifies the number of group times between decrementing the rcXformScale factor at the beginning of a slice (see section 6.8.2 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
14:5			Reserved Field: Yes
4:0	rb_first_line_bpg_offset_r	R/W	Specifies the number of additional bits that are allocated for each group on the first line of a slice. Value After Reset: 0x0 Exists: Always

5.1.103 DSC_ENC_PPS_28_31

- **Name:** Picture Parameter Set[28,31].
- **Description:** Holds the bytes in the range [28,31] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x27c
- **Exists:** DSI_HOST_DSC_ENC==1

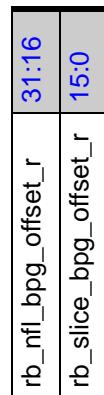


Table 5-107 Fields for Register: DSC_ENC_PPS_28_31

Bits	Name	Memory Access	Description
31:16	rb_nfl_bpg_offset_r	R/W	<p>Specifies the number of bits (including fractional bits) that are de-allocated for each group, for groups after the first line of a slice. If the first line has an additional bit budget, the additional bits that are allocated must come out of the budget for coding the remainder of the slice. Therefore, the value must be programmed to $\text{first_line_bpg_offset} / (\text{slice_height} - 1)$, then rounded up to 16 fractional bits..</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
15:0	rb_slice_bpg_offset_r	R/W	<p>Specifies the number of bits (including fractional bits) that are de-allocated for each group to enforce the slice constraint (i.e., the final buffer model fullness cannot exceed the initial transmission delay times bits per group), while allowing a programmable initial_offset. If the initial rate control (RC) model condition is not completely full, the difference between the initial RC model offset and size (initial_offset and rc_model_size_r, respectively) must be accounted for. The slice_bpg_offset parameter provides a means to resolve this difference. This parameter also allows the RC algorithm to account for bits that might be lost to SSM at the end of a slice. The value must be programmed to $(rc_model_size_r - initial_offset + numExtraMuxBits) / groupsTotal$, then rounded up to 16 fractional bits. (numExtraMuxBits and groupsTotal are described in Table E-1 of the VESA DSC Standard v1.1.)</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.104 DSC_ENC_PPS_32_35

- **Name:** Picture Parameter Set[32,35].
- **Description:** Holds the bytes in the range [32,35] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x280
- **Exists:** DSI_HOST_DSC_ENC==1



Table 5-108 Fields for Register: DSC_ENC_PPS_32_35

Bits	Name	Memory Access	Description
31:16	rb_initial_offset_r	R/W	<p>Specifies the initial value for rcXformOffset , which is initial_offset - rc_model_size_r at the start of a slice (see section 6.8.2 of VESA DSC Standard v1.1).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:0	rb_final_offset_r	R/W	<p>Specifies the maximum end-of-slice value for rcXformOffset , which is final_offset - rc_model_size_r (see section 6.8.2 of VESA DSC Standard v1.1). To ensure HRD compliance, the final_offset parameter value must be equal to rc_model_size_r - initial_xmit_delay * bits_per_pixel + numExtraMuxBits. (numExtraMuxBits is described in Table E-1).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.105 DSC_ENC_PPS_36_39

- **Name:** Picture Parameter Set[36,39].
- **Description:** Holds the bytes in the range [36,39] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x284
- **Exists:** DSI_HOST_DSC_ENC==1

reserved_31_29	31:29
rb_flatness_min_qp_r	28:24
Rsvd	23:21
rb_flatness_max_qp_r	20:16
rc_model_size_r	15:0

Table 5-109 Fields for Register: DSC_ENC_PPS_36_39

Bits	Name	Memory Access	Description
31:29	reserved_31_29	R/W	Reserved and read as zero. Value After Reset: 0x0 Exists: Always
28:24	rb_flatness_min_qp_r	R/W	Specifies the minimum QP at which flatness is signaled and the flatness QP adjustment is made. Value After Reset: 0x0 Exists: Always
23:21			Reserved Field: Yes
20:16	rb_flatness_max_qp_r	R/W	Specifies the maximum QP at which flatness is signaled and the flatness QP adjustment is made. Value After Reset: 0x0 Exists: Always
15:0	rc_model_size_r	R/W	Specifies the number of bits within the "RC model," which is described in section 6.8.2 of VESA DSC Standard v1.1. Value After Reset: 0x0 Exists: Always

5.1.106 DSC_ENC_PPS_40_43

- **Name:** Picture Parameter Set[40,43].
- **Description:** Holds the bytes in the range [40,43] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x288
- **Exists:** DSI_HOST_DSC_ENC==1

Rsvd	31:28
rc_edge_factor_r	27:24
Rsvd	23:21
rc_quant_incr_limit0_r	20:16
Rsvd	15:13
rc_quant_incr_limit1_r	12:8
rc_tgt_offset_hi_r	7:4
rc_tgt_offset_lo_r	3:0

Table 5-110 Fields for Register: DSC_ENC_PPS_40_43

Bits	Name	Memory Access	Description
31:28			Reserved Field: Yes
27:24	rc_edge_factor_r	R/W	Compared to the ratio of current activity versus previous activity to determine the presence of an "edge", which in turn determines whether the QP is incremented in the short-term RC (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
23:21			Reserved Field: Yes
20:16	rc_quant_incr_limit0_r	R/W	QP threshold 0 that is used in the short-term RC (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
15:13			Reserved Field: Yes
12:8	rc_quant_incr_limit1_r	R/W	QP threshold 1 that is used in the short-term RC (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
7:4	rc_tgt_offset_hi_r	R/W	<p>Specifies the upper end of the variability range around the target bits per group that is allowed by the short-term RC (see section 6.8.4 of VESA DSC Standard v1.1).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3:0	rc_tgt_offset_lo_r	R/W	<p>Specifies the lower end of the variability range around the target bits per group that is allowed by the short-term RC (see section 6.8.4 of VESA DSC Standard v1.1).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.107 DSC_ENC_PPS_44_47

- **Name:** Picture Parameter Set[44,47].
- **Description:** Holds the bytes in the range [44,47] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x28c
- **Exists:** DSI_HOST_DSC_ENC==1

rc_buf_thresh__0_r	31:24
rc_buf_thresh__1_r	23:16
rc_buf_thresh__2_r	15:8
rc_buf_thresh__3_r	7:0

Table 5-111 Fields for Register: DSC_ENC_PPS_44_47

Bits	Name	Memory Access	Description
31:24	rc_buf_thresh__0_r	R/W	<p>Threshold 0. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
23:16	rc_buf_thresh__1_r	R/W	<p>Threshold 1. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:8	rc_buf_thresh__2_r	R/W	<p>Threshold 2. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
7:0	rc_buf_thresh__3_r	R/W	<p>Threshold 3. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.108 DSC_ENC_PPS_48_51

- **Name:** Picture Parameter Set[48,51].
- **Description:** Holds the bytes in the range [48,51] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x290
- **Exists:** DSI_HOST_DSC_ENC==1

rc_buf_thresh_4_r	31:24
rc_buf_thresh_5_r	23:16
rc_buf_thresh_6_r	15:8
rc_buf_thresh_7_r	7:0

Table 5-112 Fields for Register: DSC_ENC_PPS_48_51

Bits	Name	Memory Access	Description
31:24	rc_buf_thresh_4_r	R/W	<p>Threshold 4. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
23:16	rc_buf_thresh_5_r	R/W	<p>Threshold 5. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:8	rc_buf_thresh_6_r	R/W	<p>Threshold 6. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
7:0	rc_buf_thresh__7_r	R/W	<p>Threshold 7. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.109 DSC_ENC_PPS_52_55

- **Name:** Picture Parameter Set[52,55].
- **Description:** Holds the bytes in the range [52,55] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x294
- **Exists:** DSI_HOST_DSC_ENC==1

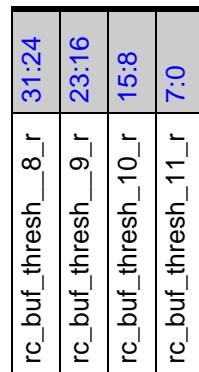


Table 5-113 Fields for Register: DSC_ENC_PPS_52_55

Bits	Name	Memory Access	Description
31:24	rc_buf_thresh_8_r	R/W	<p>Threshold 8. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
23:16	rc_buf_thresh_9_r	R/W	<p>Threshold 9. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:8	rc_buf_thresh_10_r	R/W	<p>Threshold 10. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
7:0	rc_buf_thresh_11_r	R/W	<p>Threshold 11. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.110 DSC_ENC_PPS_56_59

- **Name:** Picture Parameter Set[56,59].
- **Description:** Holds the bytes in the range [56,59] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x298
- **Exists:** DSI_HOST_DSC_ENC==1

rc_buf_thresh_12_r	31:24
rc_buf_thresh_13_r	23:16
rc_range_0_min_qp_r	15:11
rc_range_0_max_qp_r	10:6
rc_range_0_bpg_offset_r	5:0

Table 5-114 Fields for Register: DSC_ENC_PPS_56_59

Bits	Name	Memory Access	Description
31:24	rc_buf_thresh_12_r	R/W	<p>Threshold 12. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
23:16	rc_buf_thresh_13_r	R/W	<p>Threshold 13. Specify thresholds in the "RC model" for the 15 ranges defined by 14 thresholds (0 through 13, respectively) (see section 6.8.3 of VESA DSC Standard v1.1). Six 0s are appended to the lsb of each threshold value.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
15:11	rc_range_0_min_qp_r	R/W	<p>range_min_qp 0. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
10:6	rc_range__0_max_qp_r	R/W	range_max_qp 0. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
5:0	rc_range__0_bpg_offset_r	R/W	range_bpg_offset 0. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.111 DSC_ENC_PPS_60_63

- **Name:** Picture Parameter Set[60,63].
- **Description:** Holds the bytes in the range [60,63] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x29c
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range__1_min_qp_r	31:27
rc_range__1_max_qp_r	26:22
rc_range__1_bpg_offset_r	21:16
rc_range__2_min_qp_r	15:11
rc_range__2_max_qp_r	10:6
rc_range__2_bpg_offset_r	5:0

Table 5-115 Fields for Register: DSC_ENC_PPS_60_63

Bits	Name	Memory Access	Description
31:27	rc_range__1_min_qp_r	R/W	range_min_qp 1. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
26:22	rc_range__1_max_qp_r	R/W	range_max_qp 1. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
21:16	rc_range__1_bpg_offset_r	R/W	range_bpg_offset 1. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15:11	rc_range__2_min_qp_r	R/W	range_min_qp 2. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
10:6	rc_range__2_max_qp_r	R/W	range_max_qp 2. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
5:0	rc_range__2_bpg_offset_r	R/W	range_bpg_offset 2. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

5.1.112 DSC_ENC_PPS_64_67

- **Name:** Picture Parameter Set[64,67].
- **Description:** Holds the bytes in the range [64,67] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x2a0
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range_3_min_qp_r	31:27
rc_range_3_max_qp_r	26:22
rc_range_3_bpg_offset_r	21:16
rc_range_4_min_qp_r	15:11
rc_range_4_max_qp_r	10:6
rc_range_4_bpg_offset_r	5:0

Table 5-116 Fields for Register: DSC_ENC_PPS_64_67

Bits	Name	Memory Access	Description
31:27	rc_range_3_min_qp_r	R/W	range_min_qp 3. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
26:22	rc_range_3_max_qp_r	R/W	range_max_qp 3. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
21:16	rc_range_3_bpg_offset_r	R/W	range_bpg_offset 3. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15:11	rc_range__4_min_qp_r	R/W	range_min_qp 4. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
10:6	rc_range__4_max_qp_r	R/W	range_max_qp 4. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1) Value After Reset: 0x0 Exists: Always
5:0	rc_range__4_bpg_offset_r	R/W	range_bpg_offset 4. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

5.1.113 DSC_ENC_PPS_68_71

- **Name:** Picture Parameter Set[68,71].
- **Description:** Holds the bytes in the range [68,71] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x2a4
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range__5_min_qp_r	31:27
rc_range__5_max_qp_r	26:22
rc_range__5_bpg_offset_r	21:16
rc_range__6_min_qp_r	15:11
rc_range__6_max_qp_r	10:6
rc_range__6_bpg_offset_r	5:0

Table 5-117 Fields for Register: DSC_ENC_PPS_68_71

Bits	Name	Memory Access	Description
31:27	rc_range__5_min_qp_r	R/W	range_min_qp 5. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
26:22	rc_range__5_max_qp_r	R/W	range_max_qp 5. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
21:16	rc_range__5_bpg_offset_r	R/W	range_bpg_offset 5. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15:11	rc_range__6_min_qp_r	R/W	range_min_qp 6. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
10:6	rc_range__6_max_qp_r	R/W	range_max_qp 6. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always
5:0	rc_range__6_bpg_offset_r	R/W	range_bpg_offset 6. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see section 6.8.4 of VESA DSC Standard v1.1). Value After Reset: 0x0 Exists: Always

5.1.114 DSC_ENC_PPS_72_75

- **Name:** Picture Parameter Set[72,75].
- **Description:** Holds the bytes in the range [72,75] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x2a8
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range_7_min_qp_r	31:27
rc_range_7_max_qp_r	26:22
rc_range_7_bpg_offset_r	21:16
rc_range_8_min_qp_r	15:11
rc_range_8_max_qp_r	10:6
rc_range_8_bpg_offset_r	5:0

Table 5-118 Fields for Register: DSC_ENC_PPS_72_75

Bits	Name	Memory Access	Description
31:27	rc_range_7_min_qp_r	R/W	range_min_qp 7. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
26:22	rc_range_7_max_qp_r	R/W	range_max_qp 7. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
21:16	rc_range_7_bpg_offset_r	R/W	range_bpg_offset 7. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15:11	rc_range__8_min_qp_r	R/W	range_min_qp 8. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
10:6	rc_range__8_max_qp_r	R/W	range_max_qp 8. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
5:0	rc_range__8_bpg_offset_r	R/W	range_bpg_offset 8. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

5.1.115 DSC_ENC_PPS_76_79

- **Name:** Picture Parameter Set[76,79].
- **Description:** Holds the bytes in the range [76,79] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x2ac
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range__9_min_qp_r	31:27
rc_range__9_max_qp_r	26:22
rc_range__9_bpg_offset_r	21:16
rc_range__10_min_qp_r	15:11
rc_range__10_max_qp_r	10:6
rc_range__10_bpg_offset_r	5:0

Table 5-119 Fields for Register: DSC_ENC_PPS_76_79

Bits	Name	Memory Access	Description
31:27	rc_range__9_min_qp_r	R/W	range_min_qp 9. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
26:22	rc_range__9_max_qp_r	R/W	range_max_qp 9. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
21:16	rc_range__9_bpg_offset_r	R/W	range_bpg_offset 9. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15:11	rc_range_10_min_qp_r	R/W	range_min_qp 10. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
10:6	rc_range_10_max_qp_r	R/W	range_max_qp 10. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
5:0	rc_range_10_bpg_offset_r	R/W	range_bpg_offset 10. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

5.1.116 DSC_ENC_PPS_80_83

- **Name:** Picture Parameter Set[80,83].
- **Description:** Holds the bytes in the range [80,83] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x2b0
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range_11_min_qp_r	31:27
rc_range_11_max_qp_r	26:22
rc_range_11_bpg_offset_r	21:16
rc_range_12_min_qp_r	15:11
rc_range_12_max_qp_r	10:6
rc_range_12_bpg_offset_r	5:0

Table 5-120 Fields for Register: DSC_ENC_PPS_80_83

Bits	Name	Memory Access	Description
31:27	rc_range_11_min_qp_r	R/W	range_min_qp 11. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
26:22	rc_range_11_max_qp_r	R/W	range_max_qp 11. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
21:16	rc_range_11_bpg_offset_r	R/W	range_bpg_offset 11. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15:11	rc_range_12_min_qp_r	R/W	range_min_qp 12. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
10:6	rc_range_12_max_qp_r	R/W	range_max_qp 12. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
5:0	rc_range_12_bpg_offset_r	R/W	range_bpg_offset 12. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

5.1.117 DSC_ENC_PPS_84_87

- **Name:** Picture Parameter Set[84,87].
- **Description:** Holds the bytes in the range [84,87] of the Picture Parameter Set (PPS).
- **Size:** 32 bits
- **Offset:** 0x2b4
- **Exists:** DSI_HOST_DSC_ENC==1

rc_range_14_bpg_offset_r	31:27
rc_range_14_max_qp_r	26:22
rc_range_14_min_qp_r	21:16
rc_range_13_bpg_offset_r	15:11
rc_range_13_max_qp_r	10:6
rc_range_13_min_qp_r	5:0

Table 5-121 Fields for Register: DSC_ENC_PPS_84_87

Bits	Name	Memory Access	Description
31:27	rc_range_14_bpg_offset_r	R/W	<p>range_bpg_offset 14. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
26:22	rc_range_14_max_qp_r	R/W	<p>range_max_qp 14. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
21:16	rc_range_14_min_qp_r	R/W	<p>range_min_qp 14. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4).</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
15:11	rc_range_13_bpg_offset_r	R/W	range_bpg_offset 13. Specifies the target bits per group adjustment that is performed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
10:6	rc_range_13_max_qp_r	R/W	range_max_qp 13. Specifies the maximum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always
5:0	rc_range_13_min_qp_r	R/W	range_min_qp 13. Specifies the minimum QP that is allowed if the RC model has tracked to the current range (see Section 6.8.4). Value After Reset: 0x0 Exists: Always

5.1.118 INT_ST0_AP

- **Name:** Internal Diagnosis Group 0.
- **Description:** Interrupt group caused by Internal Diagnosis Group 0. Reading this register clears it.
- **Size:** 32 bits
- **Offset:** 0x300
- **Exists:** (0)==1

Rsvd	31:21
dphy_errors_4	20
dphy_errors_3	19
dphy_errors_2	18
dphy_errors_1	17
dphy_errors_0	16
ack_with_err_15	15
ack_with_err_14	14
ack_with_err_13	13
ack_with_err_12	12
ack_with_err_11	11
ack_with_err_10	10
ack_with_err_9	9
ack_with_err_8	8
ack_with_err_7	7
ack_with_err_6	6
ack_with_err_5	5
ack_with_err_4	4
ack_with_err_3	3
ack_with_err_2	2
ack_with_err_1	1
ack_with_err_0	0

Table 5-122 Fields for Register: INT_ST0_AP

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	dphy_errors_4	RC	This bit indicates LP1 contention error ErrContentionLP1 from Lane 0. Value After Reset: 0x0 Exists: Always
19	dphy_errors_3	RC	This bit indicates LP0 contention error ErrContentionLP0 from Lane 0. Value After Reset: 0x0 Exists: Always
18	dphy_errors_2	RC	This bit indicates control error ErrControl from Lane 0. Value After Reset: 0x0 Exists: Always
17	dphy_errors_1	RC	This bit indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0. Value After Reset: 0x0 Exists: Always
16	dphy_errors_0	RC	This bit indicates ErrEsc escape entry error from Lane 0. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15	ack_with_err_15	RC	<p>This bit retrieves the DSI protocol violation from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
14	ack_with_err_14	RC	<p>This bit retrieves the reserved (specific to device) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	ack_with_err_13	RC	<p>This bit retrieves the invalid transmission length from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
12	ack_with_err_12	RC	<p>This bit retrieves the DSI VC ID Invalid from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	ack_with_err_11	RC	<p>This bit retrieves the not recognized DSI data type from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	ack_with_err_10	RC	<p>This bit retrieves the checksum error (long packet only) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	ack_with_err_9	RC	<p>This bit retrieves the ECC error, multi-bit (detected, not corrected) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	ack_with_err_8	RC	<p>This bit retrieves the ECC error, single-bit (detected and corrected) from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	ack_with_err_7	RC	<p>This bit retrieves the reserved (specific to device) from the acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
6	ack_with_err_6	RC	<p>This bit retrieves the False Control error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5	ack_with_err_5	RC	<p>This bit retrieves the Peripheral Timeout error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	ack_with_err_4	RC	<p>This bit retrieves the low-power Transmit Sync error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	ack_with_err_3	RC	<p>This bit retrieves the Escape Mode Entry Command error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	ack_with_err_2	RC	<p>This bit retrieves the EoT Sync error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1	ack_with_err_1	RC	<p>This bit retrieves the SoT Sync error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
0	ack_with_err_0	RC	<p>This bit retrieves the SoT error from the Acknowledge error report.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.119 INT_MSK0_AP

- **Name:** Mask for INT_ST0_AP.
- **Description:** This register configures masks for the sources of interrupts that affect the INT_ST0_AP register. Write 1 to un-mask each error report.
- **Size:** 32 bits
- **Offset:** 0x304
- **Exists:** (0)==1

Rsvd	31:21
mask_dphy_errors_4	20
mask_dphy_errors_3	19
mask_dphy_errors_2	18
mask_dphy_errors_1	17
mask_dphy_errors_0	16
mask_ack_with_err_15	15
mask_ack_with_err_14	14
mask_ack_with_err_13	13
mask_ack_with_err_12	12
mask_ack_with_err_11	11
mask_ack_with_err_10	10
mask_ack_with_err_9	9
mask_ack_with_err_8	8
mask_ack_with_err_7	7
mask_ack_with_err_6	6
mask_ack_with_err_5	5
mask_ack_with_err_4	4
mask_ack_with_err_3	3
mask_ack_with_err_2	2
mask_ack_with_err_1	1
mask_ack_with_err_0	0

Table 5-123 Fields for Register: INT_MSK0_AP

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	mask_dphy_errors_4	R/W	Mask for dphy_errors_4 Value After Reset: 0x0 Exists: Always
19	mask_dphy_errors_3	R/W	Mask for dphy_errors_3 Value After Reset: 0x0 Exists: Always
18	mask_dphy_errors_2	R/W	Mask for dphy_errors_2 Value After Reset: 0x0 Exists: Always
17	mask_dphy_errors_1	R/W	Mask for dphy_errors_1 Value After Reset: 0x0 Exists: Always
16	mask_dphy_errors_0	R/W	Mask for dphy_errors_0 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15	mask_ack_with_err_15	R/W	Mask for ack_with_err_15 Value After Reset: 0x0 Exists: Always
14	mask_ack_with_err_14	R/W	Mask for ack_with_err_14 Value After Reset: 0x0 Exists: Always
13	mask_ack_with_err_13	R/W	Mask for ack_with_err_13 Value After Reset: 0x0 Exists: Always
12	mask_ack_with_err_12	R/W	Mask for ack_with_err_12 Value After Reset: 0x0 Exists: Always
11	mask_ack_with_err_11	R/W	Mask for ack_with_err_11 Value After Reset: 0x0 Exists: Always
10	mask_ack_with_err_10	R/W	Mask for ack_with_err_10 Value After Reset: 0x0 Exists: Always
9	mask_ack_with_err_9	R/W	Mask for ack_with_err_9 Value After Reset: 0x0 Exists: Always
8	mask_ack_with_err_8	R/W	Mask for ack_with_err_8 Value After Reset: 0x0 Exists: Always
7	mask_ack_with_err_7	R/W	Mask for ack_with_err_7 Value After Reset: 0x0 Exists: Always
6	mask_ack_with_err_6	R/W	Mask for ack_with_err_6 Value After Reset: 0x0 Exists: Always
5	mask_ack_with_err_5	R/W	Mask for ack_with_err_5 Value After Reset: 0x0 Exists: Always
4	mask_ack_with_err_4	R/W	Mask for ack_with_err_4 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
3	mask_ack_with_err_3	R/W	Mask for ack_with_err_3 Value After Reset: 0x0 Exists: Always
2	mask_ack_with_err_2	R/W	Mask for ack_with_err_2 Value After Reset: 0x0 Exists: Always
1	mask_ack_with_err_1	R/W	Mask for ack_with_err_1 Value After Reset: 0x0 Exists: Always
0	mask_ack_with_err_0	R/W	Mask for ack_with_err_0 Value After Reset: 0x0 Exists: Always

5.1.120 INT_FORCE0_AP

- **Name:** Force to INT_ST0_AP.
- **Description:** This register allows triggering interrupt events individually. Setting any bit of these register to 1 triggers the corresponding bit of INT_ST0_AP.
- **Size:** 32 bits
- **Offset:** 0x308
- **Exists:** (0)==1

Rsvd	31:21
force_dphy_errors_4	20
force_dphy_errors_3	19
force_dphy_errors_2	18
force_dphy_errors_1	17
force_dphy_errors_0	16
force_ack_with_err_15	15
force_ack_with_err_14	14
force_ack_with_err_13	13
force_ack_with_err_12	12
force_ack_with_err_11	11
force_ack_with_err_10	10
force_ack_with_err_9	9
force_ack_with_err_8	8
force_ack_with_err_7	7
force_ack_with_err_6	6
force_ack_with_err_5	5
force_ack_with_err_4	4
force_ack_with_err_3	3
force_ack_with_err_2	2
force_ack_with_err_1	1
force_ack_with_err_0	0

Table 5-124 Fields for Register: INT_FORCE0_AP

Bits	Name	Memory Access	Description
31:21			Reserved Field: Yes
20	force_dphy_errors_4	R/W	Force dphy_errors_4 Value After Reset: 0x0 Exists: Always
19	force_dphy_errors_3	R/W	Force dphy_errors_3 Value After Reset: 0x0 Exists: Always
18	force_dphy_errors_2	R/W	Force dphy_errors_2 Value After Reset: 0x0 Exists: Always
17	force_dphy_errors_1	R/W	Force dphy_errors_1 Value After Reset: 0x0 Exists: Always
16	force_dphy_errors_0	R/W	Force dphy_errors_0 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
15	force_ack_with_err_15	R/W	Force ack_with_err_15 Value After Reset: 0x0 Exists: Always
14	force_ack_with_err_14	R/W	Force ack_with_err_14 Value After Reset: 0x0 Exists: Always
13	force_ack_with_err_13	R/W	Force ack_with_err_13 Value After Reset: 0x0 Exists: Always
12	force_ack_with_err_12	R/W	Force ack_with_err_12 Value After Reset: 0x0 Exists: Always
11	force_ack_with_err_11	R/W	Force ack_with_err_11 Value After Reset: 0x0 Exists: Always
10	force_ack_with_err_10	R/W	Force ack_with_err_10 Value After Reset: 0x0 Exists: Always
9	force_ack_with_err_9	R/W	Force ack_with_err_9 Value After Reset: 0x0 Exists: Always
8	force_ack_with_err_8	R/W	Force ack_with_err_8 Value After Reset: 0x0 Exists: Always
7	force_ack_with_err_7	R/W	Force ack_with_err_7 Value After Reset: 0x0 Exists: Always
6	force_ack_with_err_6	R/W	Force ack_with_err_6 Value After Reset: 0x0 Exists: Always
5	force_ack_with_err_5	R/W	Force ack_with_err_5 Value After Reset: 0x0 Exists: Always
4	force_ack_with_err_4	R/W	Force ack_with_err_4 Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
3	force_ack_with_err_3	R/W	Force ack_with_err_3 Value After Reset: 0x0 Exists: Always
2	force_ack_with_err_2	R/W	Force ack_with_err_2 Value After Reset: 0x0 Exists: Always
1	force_ack_with_err_1	R/W	Force ack_with_err_1 Value After Reset: 0x0 Exists: Always
0	force_ack_with_err_0	R/W	Force ack_with_err_0 Value After Reset: 0x0 Exists: Always

5.1.121 INT ST1 AP

- **Name:** Internal Diagnosis Group 1.
 - **Description:** Interrupt group caused by Internal Diagnosis Group 1. Reading this register clears it.
 - **Size:** 32 bits
 - **Offset:** 0x310
 - **Exists:** (0)==1

Rsvd	31:20
dpi_buff_pld_under	19
Rsvd	18:13
gen_pld_recv_err	12
gen_pld_rd_err	11
gen_pld_send_err	10
gen_pld_wr_err	9
gen_cmd_wr_err	8
dpi_pld_wr_err	7
eopt_err	6
pkt_size_err	5
crc_err	4
ecc_multipl_err	3
ecc_single_err	2
too_ip_rx	1
to_hs_tx	0

Table 5-125 Fields for Register: INT_ST1_AP

Bits	Name	Memory Access	Description
31:20			Reserved Field: Yes
19	dpi_buff_pld_under	RC	This bit indicates that an underflow has occurred when reading payload to build DSI packet for video mode. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1
18:13			Reserved Field: Yes
12	gen_pld_recev_err	RC	This bit indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted. Value After Reset: 0x0 Exists: Always
11	gen_pld_rd_err	RC	This bit indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
10	gen_pld_send_err	RC	<p>This bit indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	gen_pld_wr_err	RC	<p>This bit indicates that the system tried to write a payload data through the Generic interface and the FIFO is full. Therefore, the payload is not written.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	gen_cmd_wr_err	RC	<p>This bit indicates that the system tried to write a command through the Generic interface and the FIFO is full. Therefore, the command is not written.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7	dpi_pld_wr_err	RC	<p>This bit indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1</p>
6	eopt_err	RC	<p>This bit indicates that the EoTp packet has not been received at the end of the incoming peripheral transmission.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
5	pkt_size_err	RC	<p>This bit indicates that the packet size error has been detected during the packet reception.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
4	crc_err	RC	<p>This bit indicates that the CRC error has been detected in the received packet payload.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
3	ecc_multpl_err	RC	<p>This bit indicates that the ECC multiple error has been detected in a received packet.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
2	ecc_single_err	RC	<p>This bit indicates that the ECC single error has been detected and corrected in a received packet.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
1	to_lp_rx	RC	This bit indicates that the low-power reception timeout counter reached the end and contention has been detected. Value After Reset: 0x0 Exists: Always
0	to_hs_tx	RC	This bit indicates that the high-speed transmission timeout counter reached the end and contention has been detected. Value After Reset: 0x0 Exists: Always

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

5.1.122 INT_MSK1_AP

- **Name:** Mask for INT_ST1_AP.
- **Description:** This register configures masks for the sources of interrupts that affect the INT_ST1_AP register. Write 1 to un-mask each error report.
- **Size:** 32 bits
- **Offset:** 0x314
- **Exists:** (0)==1

	31:20	19	18:13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsvd																
mask_dpi_buff_pld_under																
Rsvd																
mask_gen_pld_recv_err																
mask_gen_pld_rd_err																
mask_gen_pld_send_err																
mask_gen_pld_wr_err																
mask_gen_cmd_wr_err																
mask_dpi_pld_wr_err																
mask_eopt_err																
mask_pkt_size_err																
mask_crc_err																
mask_ecc_multl_err																
mask_ecc_single_err																
mask_to_lp_rx																
mask_to_hs_tx																

Table 5-126 Fields for Register: INT_MSK1_AP

Bits	Name	Memory Access	Description
31:20			Reserved Field: Yes
19	mask_dpi_buff_pld_under	R/W	Mask for dpi_buff_pld_under. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
18:13			Reserved Field: Yes
12	mask_gen_pld_recv_err	R/W	Mask for gen_pld_recv_err. Value After Reset: 0x0 Exists: Always
11	mask_gen_pld_rd_err	R/W	Mask for gen_pld_rd_err. Value After Reset: 0x0 Exists: Always
10	mask_gen_pld_send_err	R/W	Mask for gen_pld_send_err. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
9	mask_gen_pld_wr_err	R/W	Mask for gen_pld_wr_err. Value After Reset: 0x0 Exists: Always
8	mask_gen_cmd_wr_err	R/W	Mask for gen_cmd_wr_err. Value After Reset: 0x0 Exists: Always
7	mask_dpi_pld_wr_err	R/W	Mask for dpi_pld_wr_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1
6	mask_eopt_err	R/W	Mask for eopt_err. Value After Reset: 0x0 Exists: Always
5	mask_pkt_size_err	R/W	Mask for pkt_size_err. Value After Reset: 0x0 Exists: Always
4	mask_crc_err	R/W	Mask for crc_err. Value After Reset: 0x0 Exists: Always
3	mask_ecc_multpl_err	R/W	Mask for ecc_multpl_err. Value After Reset: 0x0 Exists: Always
2	mask_ecc_single_err	R/W	Mask for ecc_single_err. Value After Reset: 0x0 Exists: Always
1	mask_to_lp_rx	R/W	Mask for to_lp_rx. Value After Reset: 0x0 Exists: Always
0	mask_to_hs_tx	R/W	Mask for to_hs_tx. Value After Reset: 0x0 Exists: Always

5.1.123 INT_FORCE1_AP

- **Name:** Force to INT_ST1_AP
- **Description:** This register allows triggering interrupt events individually. Setting any bit of these register to 1 triggers the corresponding bit of INT_ST1_AP.
- **Size:** 32 bits
- **Offset:** 0x318
- **Exists:** (0)==1

Rsvd	31:20	
force_dpi_buff_pld_under	19	
Rsvd	18:13	
force_gen_pld_recev_err	12	
force_gen_pld_rd_err	11	
force_gen_pld_send_err	10	
force_gen_pld_wr_err	9	
force_gen_cmd_wr_err	8	
force_dpi_pld_wr_err	7	
force_eopt_err	6	
force_pkt_size_err	5	
force_crc_err	4	
force_ecc_multipart_err	3	
force_ecc_single_err	2	
force_to_lp_rx	1	
force_to_hs_tx	0	

Table 5-127 Fields for Register: INT_FORCE1_AP

Bits	Name	Memory Access	Description
31:20			Reserved Field: Yes
19	force_dpi_buff_pld_under	R/W	Force for dpi_buff_pld_under. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
18:13			Reserved Field: Yes
12	force_gen_pld_recev_err	R/W	Force for gen_pld_recev_err. Value After Reset: 0x0 Exists: Always
11	force_gen_pld_rd_err	R/W	Force for gen_pld_rd_err. Value After Reset: 0x0 Exists: Always
10	force_gen_pld_send_err	R/W	Force for gen_pld_send_err. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
9	force_gen_pld_wr_err	R/W	Force for gen_pld_wr_err. Value After Reset: 0x0 Exists: Always
8	force_gen_cmd_wr_err	R/W	Force for gen_cmd_wr_err. Value After Reset: 0x0 Exists: Always
7	force_dpi_pld_wr_err	R/W	Force for dpi_pld_wr_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1
6	force_eopt_err	R/W	Force for eopt_err. Value After Reset: 0x0 Exists: Always
5	force_pkt_size_err	R/W	Force for pkt_size_err. Value After Reset: 0x0 Exists: Always
4	force_crc_err	R/W	Force for crc_err. Value After Reset: 0x0 Exists: Always
3	force_ecc_multpl_err	R/W	Force for ecc_multpl_err. Value After Reset: 0x0 Exists: Always
2	force_ecc_single_err	R/W	Force for ecc_single_err. Value After Reset: 0x0 Exists: Always
1	force_to_lp_rx	R/W	Force for to_lp_rx. Value After Reset: 0x0 Exists: Always
0	force_to_hs_tx	R/W	Force for to_hs_tx. Value After Reset: 0x0 Exists: Always

5.1.124 INT_ST2_AP

- **Name:** Internal Diagnosis Group 2.
- **Description:** Interrupt group caused by Internal Diagnosis Group 2. Reading this register clears it.
- **Size:** 32 bits
- **Offset:** 0x320
- **Exists:** (0)==1

Rsvd	31:27	
bcm_dpclk_ap_err	26	
bcm_lanebyteclk_ap_err	25	
bcm_pclk_ap_err	24	
Rsvd	23:22	
genericreadmem_ecc_multpl_err	21	
genericpldmem_ecc_multpl_err	20	
dpipixelmem_ecc_multpl_err	19	
genericreadmem_ecc_err	18	
genericpldmem_ecc_err	17	
dpipixelmem_ecc_err	16	
Rsvd	15	
to_bt2_err	14	
to_lptxjpls_err	13	
to_lptxtrig_err	12	
to_lptxrdy_err	11	
to_hstxrdy_err	10	
to_lpx_err	9	
to_hstx_err	8	
Rsvd	7:3	
regbank_reg_parity_err	2	
edpi_in_parity_err	1	
apb_in_parity_err	0	

Table 5-128 Fields for Register: INT_ST2_AP

Bits	Name	Memory Access	Description
31:27			Reserved Field: Yes
26	bcm_dpclk_ap_err	RC	This bit indicates that an error occurred in BCM while synchronizing for dpclk. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
25	bcm_lanebyteclk_ap_err	RC	This bit indicates that an error occurred in BCM while synchronizing for lanebyteclk. Value After Reset: 0x0 Exists: Always
24	bcm_pclk_ap_err	RC	This bit indicates that an error occurred in BCM while synchronizing for pclk. Value After Reset: 0x0 Exists: Always
23:22			Reserved Field: Yes

Bits	Name	Memory Access	Description
21	genericreadmem_ecc_multpl_err	RC	<p>This bit indicates that an multiple ecc error occurred in Generic Read Interface 2-Port RAM.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
20	genericpldmem_ecc_multpl_err	RC	<p>This bit indicates that an multiple ecc error occurred in Generic Write Interface 2-Port RAM.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
19	dipipixelmem_ecc_multpl_err	RC	<p>This bit indicates that an multiple ecc error occurred in DPI Interface 2-Port RAM.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1</p>
18	genericreadmem_ecc_err	RC	<p>This bit indicates that an ecc error occurred in Generic Read Interface 2-Port RAM.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
17	genericpldmem_ecc_err	RC	<p>This bit indicates that an ecc error occurred in Generic Write Interface 2-Port RAM.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
16	dipipixelmem_ecc_err	RC	<p>This bit indicates that an ecc error occurred in DPI Interface 2-Port RAM.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1</p>
15			Reserved Field: Yes
14	to_bta_err	RC	<p>This bit indicates that the configured timeout counter to_bta_cfg for the low-power direction was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
13	to_lptxulps_err	RC	<p>This bit indicates that the configured timeout counter to_lptxulps_cfg for the low-power ulpsActiveNot was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Bits	Name	Memory Access	Description
12	to_lptxtrig_err	RC	<p>This bit indicates that the configured timeout counter to_lptxtrig_cfg for the low-power StopState was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
11	to_lptxrdy_err	RC	<p>This bit indicates that the configured timeout counter to_lptxrdy_cfg for the low-power TxReadyEsc was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
10	to_hstxrdy_err	RC	<p>This bit indicates that the configured timeout counter to_hstxrdy_cfg for the high-speed TxReadyHS was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
9	to_lprx_err	RC	<p>This bit indicates that the configured timeout counter to_lprx_cfg for low-power reception was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
8	to_hstx_err	RC	<p>This bit indicates that the configured timeout counter to_hstx_cfg for high-speed transmission was reached.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
7:3			Reserved Field: Yes
2	regbank_reg_parity_err	RC	<p>This bit reports a parity error in configuration registers.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
1	edpi_in_parity_err	RC	<p>This bit reports an eDPI interface parity error.</p> <p>Value After Reset: 0x0</p> <p>Exists: (DSI_HOST_DATAINTERFACE == 4)==1</p>
0	apb_in_parity_err	RC	<p>This bit reports an APB interface parity error.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.125 INT MSK2 AP

- **Name:** Mask for INT_ST2_AP.
 - **Description:** This register configures masks for the sources of interrupts that affect the INT_ST2_AP register. Write 1 to un-mask each error report.
 - **Size:** 32 bits
 - **Offset:** 0x324
 - **Exists:** (0)==1

Rsvd		31:27
mask_bcm_dpclk_ap_err	26	25
mask_bcm_pcik_ap_err	24	24
Rsvd		23:22
mask_genericreadmem_ecc_multpl_err	21	21
mask_genERICpldmem_ecc_multpl_err	20	20
mask_dpipixelmem_ecc_multpl_err	19	19
mask_genericreadmem_ecc_err	18	18
mask_genERICpldmem_ecc_err	17	17
mask_dpipixelmem_ecc_err	16	16
Rsvd		15
mask_to_btاء_err	14	14
mask_to_lptxulpس_err	13	13
mask_to_lptxtrig_err	12	12
mask_to_lptxrdy_err	11	11
mask_to_hstxrdy_err	10	10
mask_to_lprx_err	9	9
mask_to_hstx_err	8	8
Rsvd		7:3
mask_Regbank_Reg_Parity_err	2	2
mask_edpi_in_parity_err	1	1
mask_app_in_parity_err	0	0

Table 5-129 Fields for Register: INT_MSK2_AP

Bits	Name	Memory Access	Description
31:27			Reserved Field: Yes
26	mask_bcm_dpclk_ap_err	R/W	<p>Mask for bcm_dpclk_ap_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1</p>
25	mask_bcm_lanebyteclk_ap_err	R/W	<p>Mask for bcm_lanebyteclk_ap_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
24	mask_bcm_pclk_ap_err	R/W	<p>Mask for bcm_pclk_ap_err.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>
23:22			Reserved Field: Yes

Bits	Name	Memory Access	Description
21	mask_genericreadmem_ecc_multpl_err	R/W	Mask for genericreadmem_ecc_multpl_err. Value After Reset: 0x0 Exists: Always
20	mask_genericpldmem_ecc_multpl_err	R/W	Mask for genericpldmem_ecc_multpl_err. Value After Reset: 0x0 Exists: Always
19	mask_dpipixelmem_ecc_multpl_err	R/W	Mask for dpipixelmem_ecc_multpl_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
18	mask_genericreadmem_ecc_err	R/W	Mask for genericreadmem_ecc_err. Value After Reset: 0x0 Exists: Always
17	mask_genericpldmem_ecc_err	R/W	Mask for genericpldmem_ecc_err. Value After Reset: 0x0 Exists: Always
16	mask_dpipixelmem_ecc_err	R/W	Mask for dpipixelmem_ecc_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
15			Reserved Field: Yes
14	mask_to_bta_err	R/W	Mask for to_bta_err. Value After Reset: 0x0 Exists: Always
13	mask_to_lptxulp_err	R/W	Mask for to_lptxulp_err. Value After Reset: 0x0 Exists: Always
12	mask_to_lptxtrig_err	R/W	Mask for to_lptxtrig_err. Value After Reset: 0x0 Exists: Always
11	mask_to_lptxrdy_err	R/W	Mask for to_lptxrdy_err. Value After Reset: 0x0 Exists: Always
10	mask_to_hstxrdy_err	R/W	Mask for to_hstxrdy_err. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
9	mask_to_lprx_err	R/W	Mask for to_lprx_err. Value After Reset: 0x0 Exists: Always
8	mask_to_hstx_err	R/W	Mask for to_hstx_err. Value After Reset: 0x0 Exists: Always
7:3			Reserved Field: Yes
2	mask_regbank_reg_parity_err	R/W	Mask for regbank_reg_parity_err. Value After Reset: 0x0 Exists: Always
1	mask_edpi_in_parity_err	R/W	Mask for edpi_in_parity_err. Value After Reset: 0x0 Exists: (DSI_HOST_DATAINTERFACE == 4)==1
0	mask_apb_in_parity_err	R/W	Mask for apb_in_parity_err. Value After Reset: 0x0 Exists: Always

5.1.126 INT_FORCE2_AP

- **Name:** Force to INT_ST2_AP.
- **Description:** This register allows triggering interrupt events individually. Setting any bit of these register to 1 triggers the corresponding bit of INT_ST2_AP.
- **Size:** 32 bits
- **Offset:** 0x328
- **Exists:** (0)==1

Rsvd	31:27	
force_bcm_dpclk_ap_err	26	
force_bcm_lanebyteclk_ap_err	25	
force_bcm_pclk_ap_err	24	
Rsvd	23:22	
force_genericreadmem_ecc_multpl_err	21	
force_genericpldmem_ecc_multpl_err	20	
force_dpipixmem_ecc_multpl_err	19	
force_genericreadmem_ecc_err	18	
force_genericpldmem_ecc_err	17	
force_dpipixmem_ecc_err	16	
Rsvd	15	
force_to_bt8_err	14	
force_to_lptxulp8_err	13	
force_to_lptxtrig_err	12	
force_to_lptxrdy_err	11	
force_to_hstxrdy_err	10	
force_to_lprx_err	9	
force_to_hstx_err	8	
Rsvd	7:3	
force_regbank_reg_parity_err	2	
force_edpi_in_parity_err	1	
force_app_in_parity_err	0	

Table 5-130 Fields for Register: INT_FORCE2_AP

Bits	Name	Memory Access	Description
31:27			Reserved Field: Yes
26	force_bcm_dpclk_ap_err	R/W	Force for bcm_dpclk_ap_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))==1
25	force_bcm_lanebyteclk_ap_err	R/W	Force for bcm_lanebyteclk_ap_err. Value After Reset: 0x0 Exists: Always
24	force_bcm_pclk_ap_err	R/W	Force for bcm_pclk_ap_err. Value After Reset: 0x0 Exists: Always
23:22			Reserved Field: Yes

Bits	Name	Memory Access	Description
21	force_genericreadmem_ecc_multpl_err	R/W	Force for genericreadmem_ecc_multpl_err. Value After Reset: 0x0 Exists: Always
20	force_genericpldmem_ecc_multpl_err	R/W	Force for genericpldmem_ecc_multpl_err. Value After Reset: 0x0 Exists: Always
19	force_dpipixelmem_ecc_multpl_err	R/W	Force for dpipixelmem_ecc_multpl_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1
18	force_genericreadmem_ecc_err	R/W	Force for genericreadmem_ecc_err. Value After Reset: 0x0 Exists: Always
17	force_genericpldmem_ecc_err	R/W	Force for genericpldmem_ecc_err. Value After Reset: 0x0 Exists: Always
16	force_dpipixelmem_ecc_err	R/W	Force for dpipixelmem_ecc_err. Value After Reset: 0x0 Exists: ((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4)) == 1
15			Reserved Field: Yes
14	force_to_bta_err	R/W	Mas for to_bta_err. Value After Reset: 0x0 Exists: Always
13	force_to_lptxulp_err	R/W	Force for to_lptxulp_err. Value After Reset: 0x0 Exists: Always
12	force_to_lptxtrig_err	R/W	Force for to_lptxtrig_err. Value After Reset: 0x0 Exists: Always
11	force_to_lptxrdy_err	R/W	Force for to_lptxrdy_err. Value After Reset: 0x0 Exists: Always
10	force_to_hstxrdy_err	R/W	Force for to_hstxrdy_err. Value After Reset: 0x0 Exists: Always

Bits	Name	Memory Access	Description
9	force_to_lprx_err	R/W	Force for to_lprx_err. Value After Reset: 0x0 Exists: Always
8	force_to_hstx_err	R/W	Force for to_hstx_err. Value After Reset: 0x0 Exists: Always
7:3			Reserved Field: Yes
2	force_regbank_reg_parity_err	R/W	Force for regbank_reg_parity_err. Value After Reset: 0x0 Exists: Always
1	force_edpi_in_parity_err	R/W	Force for edpi_in_parity_err. Value After Reset: 0x0 Exists: (DSI_HOST_DATAINTERFACE == 4)==1
0	force_apb_in_parity_err	R/W	Force for apb_in_parity_err. Value After Reset: 0x0 Exists: Always

5.1.127 TO_HSTXRDY_CFG_AP

- **Name:** HS TX RDY timeout.
- **Description:** Configures the counter for HS TX RDY timeout.
- **Size:** 32 bits
- **Offset:** 0x340
- **Exists:** (0)==1

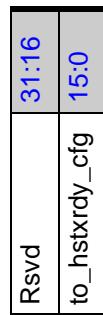


Table 5-131 Fields for Register: TO_HSTXRDY_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_hstxrdy_cfg	R/W	This field configures the counter that triggers a high-speed transmission ready timeout. Value After Reset: 0x0 Exists: Always

5.1.128 TO_LPTXRDY_CFG_AP

- **Name:** LP TX RDY timeout
- **Description:** Configures the counter for LP TX RDY timeout.
- **Size:** 32 bits
- **Offset:** 0x344
- **Exists:** (0)==1

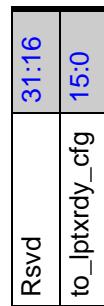


Table 5-132 Fields for Register: TO_LPTXRDY_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_lptxrdy_cfg	R/W	This field configures the counter that triggers a low-power data transmission timeout. Value After Reset: 0x0 Exists: Always

5.1.129 TO_LPTXTRIG_CFG_AP

- **Name:** LP TX TRIG timeout.
- **Description:** Configures the counter for LP TX TRIG timeout.
- **Size:** 32 bits
- **Offset:** 0x348
- **Exists:** (0)==1

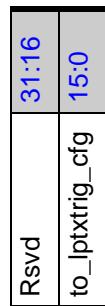


Table 5-133 Fields for Register: TO_LPTXTRIG_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_lptxtrig_cfg	R/W	This field configures the counter that triggers a low-power trigger transmission timeout. Value After Reset: 0x0 Exists: Always

5.1.130 TO_LPTXULPS_CFG_AP

- **Name:** LP TX ULPS timeout.
- **Description:** Configures the counter for LP TX ULPS timeout.
- **Size:** 32 bits
- **Offset:** 0x34c
- **Exists:** (0)==1

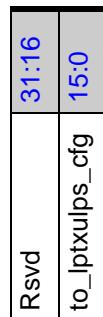


Table 5-134 Fields for Register: TO_LPTXULPS_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_lptxulps_cfg	R/W	This field configures the counter that triggers a low-power ULPS entry timeout. Value After Reset: 0x0 Exists: Always

5.1.131 TO_HSTX_CFG_AP

- **Name:** HS TX timeout.
- **Description:** Configures the counter for HS TX timeout.
- **Size:** 32 bits
- **Offset:** 0x350
- **Exists:** (0)==1

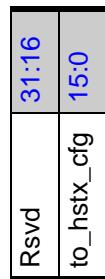


Table 5-135 Fields for Register: TO_HSTX_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_hstx_cfg	R/W	This field configures the counter that triggers a high speed transmission timeout. Value After Reset: 0x0 Exists: Always

5.1.132 TO_LPRX_CFG_AP

- **Name:** LP RX timeout.
- **Description:** Configures the counter for LP RX timeout.
- **Size:** 32 bits
- **Offset:** 0x354
- **Exists:** (0)==1

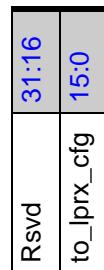


Table 5-136 Fields for Register: TO_LPRX_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_lprx_cfg	R/W	<p>This field configures the counter that triggers a low power reception timeout.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.133 TO_BTA_CFG_AP

- **Name:** Counter for BTA timeout.
- **Description:** Configures the counter for BTA timeout.
- **Size:** 32 bits
- **Offset:** 0x358
- **Exists:** (0)==1

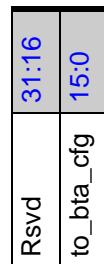


Table 5-137 Fields for Register: TO_BTA_CFG_AP

Bits	Name	Memory Access	Description
31:16			Reserved Field: Yes
15:0	to_bta_cfg	R/W	This field configures the counter that triggers a peripheral response timeout after bus turnaround completion. Value After Reset: 0x0 Exists: Always

5.1.134 TO_CLK_DIV_AP

- **Name:** Factor for internal clock dividers.
- **Description:** This register configures the factor for internal dividers to divide pclk for timeout purposes.
- **Size:** 32 bits
- **Offset:** 0x35c
- **Exists:** (0)==1



Table 5-138 Fields for Register: TO_CLK_DIV_AP

Bits	Name	Memory Access	Description
31:8			Reserved Field: Yes
7:0	to_clk_division	R/W	<p>This field indicates the division factor for pclk, used to evaluate timeouts related with DPHY. The values 0 and 1 disables the clock divider generation.</p> <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

5.1.135 ERR_INJ_CTRL_AP

- **Name:** FMEDA memory error injection.
- **Description:** This register configures memory error injection for FMEDA mode.
- **Size:** 32 bits
- **Offset:** 0x380
- **Exists:** (0)==1

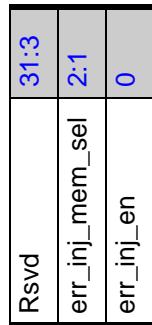


Table 5-139 Fields for Register: ERR_INJ_CTRL_AP

Bits	Name	Memory Access	Description
31:3			Reserved Field: Yes
2:1	err_inj_mem_sel	R/W	<p>This field selects particular memory interface to inject the errors: Values:</p> <ul style="list-style-type: none"> ■ 0x1 (DRAM): DPI RAM ■ 0x2 (GWRAM): Generic Write RAM ■ 0x3 (GRRAM): Generic Read RAM <p>Value After Reset: 0x0 Exists: Always</p>
0	err_inj_en	R/W	<p>This field enables the memory error injection. Value After Reset: 0x0 Exists: Always</p>

5.1.136 ERR_INJ_CHK_MSK_AP

- **Name:** Mask for FMEDA ECC check bits.
- **Description:** This register configures the error injection mask for ECC check bits in FMEDA mode.
- **Size:** 32 bits
- **Offset:** 0x384
- **Exists:** (0)==1

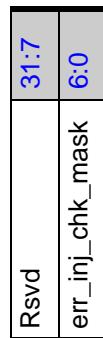


Table 5-140 Fields for Register: ERR_INJ_CHK_MSK_AP

Bits	Name	Memory Access	Description
31:7			Reserved Field: Yes
6:0	err_inj_chk_mask	R/W	Error injection mask in check bits. Value After Reset: 0x0 Exists: Always

5.1.137 ERR_INJ_DATA_MSK_AP

- **Name:** Mask for FMEDA 32-bit data error injection.
- **Description:** This register configures the error injection mask for 32 data bits in FMEDA mode.
- **Size:** 32 bits
- **Offset:** 0x388
- **Exists:** (0)==1



Table 5-141 Fields for Register: ERR_INJ_DATA_MSK_AP

Bits	Name	Memory Access	Description
31:0	err_inj_data_mask	R/W	Error injection mask in 32 data bits. Value After Reset: 0x0 Exists: Always

5.1.138 ERR_INJ_ST_AP

- **Name:** FMEDA mode error injection status.
- **Description:** This register reports error injection status for FMEDA mode.
- **Size:** 32 bits
- **Offset:** 0x38c
- **Exists:** (0)==1

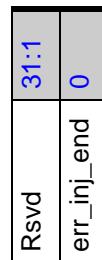


Table 5-142 Fields for Register: ERR_INJ_ST_AP

Bits	Name	Memory Access	Description
31:1			Reserved Field: Yes
0	err_inj_end	R	<p>Memory error Inject start status (autoclear):</p> <p>Values:</p> <ul style="list-style-type: none"> ■ 0x0 (ERRINJ): Indicates the memory injection has not ended or err_inj_en is low ■ 0x1 (ERRINJEND): Indicates the memory injection has ended (err_inj_en is high) <p>Value After Reset: 0x0</p> <p>Exists: Always</p>

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

A

Area and Power

Tables [Table A-1](#) through [Table A-5](#) shows the power consumption values and the area of the DWC_mipi_dsi_host configured for two data lanes and four data lanes with different interface options and support for generic packets for industry standard 7ff, 16ff, 28nm and 40nm respectively.

The D-PHY or 2-Port RAM areas are not considered in these numbers. The area is represented by two-input NAND gates.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not distribute.

Table A-1 Industry Standard 7ff

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder **	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY (Gspss) ComboPHY *†	Max System Interface frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI/ eDPI (MHz)
D-PHY	2 data lanes	DBI	-	720p	-	18	3,49	2,5	500	500	NA
			-	720p	-	30	4,46	2,5	500	NA	500
			-	720p	-	35	5,56	2,5	500	500	500
D-PHY	2 data lanes	EDPI	-	720p	-	37	5,70	2,5	500	NA	500
			-		Yes	50	9,54	2,5	500	NA	500
	4 data lanes	EDPI	-	720p	-	18	3,51	2,5	500	500	NA
			-	720p	-	31	4,48	2,5	500	NA	500
			-	720p	-	36	5,59	2,5	500	500	500
			-	720p	-	39	5,75	2,5	500	NA	500
			Single DSC	720p	-	273	3,63	2,5	500	NA	498
				4K	-	275	3,66	2,5	500	NA	500
			Dual DSC	720p	-	433	3,81	2,5	500	NA	500
				4K	-	436	3,81	2,5	500	NA	500
			Dual DSC, Dual Pixel	720p	-	513	6,42	2,5	500	NA	474
				4K	-	507	6,44	2,5	500	NA	481
			Dual DSC, Dual Port	720p	-	504	6,13	2,5	500	NA	483
				4K	-	498	6,13	2,5	500	NA	478

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder **	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY (Gsps) ComboPHY *†	Max System Interface frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI/ eDPI (MHz)
Combo D-PHY/C-PHY	D-PHY 4 data lanes / C-PHY 3 data lanes	eDPI	-	720p	-	50	7,22	2,5	500	NA	500
			Single DSC	720p	-	291	4,19	2,5	500	NA	500
				4K	-	294	4,24	2,5	500	NA	500
			Dual DSC	720p	-	455	4,27	2,5	500	NA	500
				4K	-	458	4,30	2,5	500	NA	500
			Dual DSC, Dual Port	720p	-	521	6,63	2,5	500	NA	467
				4K	-	519	6,59	2,5	500	NA	481

* Interface speed is limited according to D-PHY v1.2 and C-PHY v1.0 specifications.
 † Tests were performed using a 500Mbps/MspS step.
 ** Assuming that 7 compressed lines are stored in the Rate Buffer.

Table A-2 Industry Standard 16ffc

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY (Gsps) Combo PHY *†	Max System Interface frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI / eDPI (MHz)
D-PHY	2 data lanes	DBI	-	720p	-	14	6,22	2,5	500	500	NA
		DPI	-	720p	-	27	7,88	2,5	500	NA	500
		DBI and DPI	-	720p	-	31	9,90	2,5	500	500	500
		EDPI	-	720p	-	34	10,10	2,5	500	NA	500
	4 data lanes				Yes	44	16,90	2,5	500	NA	500
	DBI	-	720p	-	14	6,30	2,5	500	500	NA	
	DPI	-	720p	-	29	7,96	2,5	500	NA	500	
	DBI and DPI	-	720p	-	34	9,92	2,5	500	500	500	
EDPI	EDPI	-	720p	-	36	10,20	2,5	500	NA	500	
		Single DSC	720p	-	272	5,99	2,5	500	NA	282	
					274	5,90	2,5	500	NA	281	
		Dual DSC	720p	-	450	6,15	2,5	500	NA	500	
					452	6,20	2,5	500	NA	500	
		Dual DSC, Dual Pixel	720p	-	511	10,50	2,5	500	NA	275	
					511	10,50	2,5	500	NA	277	
		Dual DSC, Dual Port	720p	-	499	10,10	2,5	500	NA	277	
					502	10,00	2,5	500	NA	274	

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY (Gsps) Combo PHY *†		Max System Interface frequency (MHz)		
								APB (MHz)	DBI (MHz)	DPI / eDPI (MHz)		
Combo D-PHY/C-PHY	C-PHY 3 data lanes	EDPI	-	720p	-	48	13,00	2,5		500	NA	500
			Single DSC	720p	-	296	6,61	2,5		500	NA	288
				4K	-	300	6,67	2,5		500	NA	288
			Dual DSC	720p	-	478	6,88	2,5		500	NA	500
				4K	-	479	6,91	2,5		500	NA	500
			Dual DSC, Dual Port	720p	-	522	10,70	2,5		500	NA	277
				4K	-	521	10,70	2,5		500	NA	275

* Interface speed is limited according to D-PHY v1.2 and C-PHY v1.0 specifications.
 † Tests were performed using a 500Mbps/Msp step.
 ** Assuming that 7 compressed lines are stored in the Rate Buffer.

Table A-3 Industry Standard 28 High Performance

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps)		Max System Interface Frequency (MHz)		
								D-PHY(Gbps)	ComboPHY *†	APB (MHz)	DBI (MHz)	DPI/eDPI (MHz)
D-PHY	2 data lanes	DBI	-	720p	-	15	8,33	2,5		500	500	NA
			-	720p	-	27	10,70	2,5		500	NA	500
		DBI and DPI	-	720p	-	31	13,30	2,5		500	500	500
		EDPI	-	720p	-	33	13,70	2,5		500	NA	500
			-		Yes	45	22,90	2,5		500	NA	500
	4 data lanes	DBI	-	720p	-	15	8,40	2,5		500	500	NA
			-	720p	-	28	10,70	2,5		500	NA	500
		DBI and DPI	-	720p	-	32	13,30	2,5		500	500	500
		Single DSC	-	720p	-	34	13,80	2,5		500	NA	500
			-	720p	-	271	8,87	2,5		500	NA	448
			4K	-	270	8,82	2,5			500	NA	442
	EDPI	Dual DSC	-	720p	-	422	8,78	2,5		500	NA	500
			4K	-	426	8,88	2,5			500	NA	500
		Dual DSC, Dual Pixel	-	720p	-	503	16,10	2,5		500	NA	427
			4K	-	497	16,10	2,5			500	NA	417
		Dual DSC, Dual Port	-	720p	-	493	15,30	2,5		500	NA	435
			4K	-	494	15,40	2,5			500	NA	431

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY(Gsp/s) ComboPHY *†	Max System Interface Frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI/eDPI (MHz)
Combo D-PHY/C- PHY	D-PHY 4 data lanes / C-PHY 3 data lanes	EDPI	-	720p	-	45	17,60	2,5	500	NA	500
			Single DSC	720p	-	294	9,80	2,5	500	NA	442
				4K	-	293	10,10	2,5	500	NA	448
			Dual DSC	720p	-	444	9,92	2,5	500	NA	500
				4K	-	449	9,93	2,5	500	NA	500
			Dual DSC, Dual Port	720p	-	518	16,60	2,5	500	NA	435
				4K	-	516	16,50	2,5	500	NA	426

* Interface speed is limited according to D-PHY v1.2 and C-PHY v1.0 specifications.
 † Tests were performed using a 500Mbps/Msp/s step.
 ** Assuming that 7 compressed lines are stored in the Rate Buffer.

Table A-4 Industry Standard 28

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY(Gbps) ComboPHY *†	Max System Interface Frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI/eDP I (MHz)
D-PHY	2 data lanes	DBI	-	720p	-	14	8,80	2,5	500	500	NA
		DPI	-	720p	-	27	11,30	2,5	500	NA	500
		DBI and DPI	-	720p	-	31	14,20	2,5	500	500	500
		EDPI	-	720p	-	34	14,50	2,5	500	NA	500
					Yes	44	25,60	2,5	500	NA	500
	4 data lanes	DBI	-	720p	-	14	8,82	2,5	500	500	NA
		DPI	-	720p	-	29	11,50	2,5	500	NA	500
		DBI and DPI	-	720p	-	33	14,00	2,5	500	500	500
		EDPI	Single DSC	720p	-	35	14,50	2,5	500	NA	500
				720p	-	283	13,00	2,5	500	NA	271
				4K	-	283	13,00	2,5	500	NA	273
			Dual DSC	720p	-	460	13,60	2,5	500	NA	500
				4K	-	460	13,60	2,5	500	NA	500
			Dual DSC, Dual Pixel	720p	-	526	22,90	2,5	500	NA	267
				4K	-	528	23,00	2,5	500	NA	268
			Dual DSC, Dual Port	720p	-	516	22,10	2,5	500	NA	267
				4K	-	516	22,20	2,5	500	NA	266

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY(Gsps) ComboPHY *†	Max System Interface Frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI/eDP I (MHz)
Combo D-PHY/C- PHY	D-PHY 4 data lanes / C-PHY 3 data lanes	EDPI	-	720p	-	48	18,70	2,5	500	NA	500
			Single DSC	720p	-	309	14,50	2,5	500	NA	273
				4K	-	307	14,50	2,5	500	NA	270
			Dual DSC	720p	-	483	14,90	2,5	500	NA	500
				4K	-	485	15,00	2,5	500	NA	500
			Dual DSC, Dual Port	720p	-	535	23,30	2,5	500	NA	267
				4K	-	536	23,40	2,5	500	NA	264

* Interface speed is limited according to D-PHY v1.2 and C-PHY v1.0 specifications.
 † Tests were performed using a 500Mbps/Msp step.
 ** Assuming that 7 compressed lines are stored in the Rate Buffer.

Table A-5 Industry Standard 40

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY (Gsp/s) ComboPHY*†	Max System Interface Frequency (MHz)			
									APB (MHz)	DBI (MHz)	DPI/eDPI (MHz)	
D-PHY	2 data lanes	DBI	-	720p	-	15	9,34	2,5	500	500	NA	
		DPI	-	720p	-	29	10,80	2,5	500	NA	500	
		DBI and DPI	-	720p	-	33	13,60	2,5	500	500	500	
		EDPI	-	720p	-	34	12,00	2,5	500	NA	200	
					Yes	45	20,80	2,5	500	NA	200	
	4 data lanes	DBI	-	720p	-	15	9,45	2,5	500	500	NA	
		DPI	-	720p	-	31	10,90	2,5	500	NA	500	
		DBI and DPI	-	720p	-	36	13,60	2,5	500	500	500	
		EDPI	Single DSC	720p	-	36	12,20	2,5	500	NA	200	
				720p	-	302	16,20	2	500	NA	107	
				4K	-	302	16,30	2	500	NA	107	
				Dual DSC	720p	-	553	17,60	2	500	NA	394
					4K	-	558	17,70	2	500	NA	395
			Dual DSC, Dual Pixel	720p	-	565	29,00	2	500	NA	104	
				4K	-	566	29,10	2	500	NA	104	
			Dual DSC, Dual Port	720p	-	554	23,40	2	500	NA	110	
				4K	-	555	27,40	2	500	NA	105	

Physical Interface	Physical Interface Configuration	System Interface	DSC Encoder	Resolution	Automotive Package	Area (Kgates)	Power [mW]	Max Physical Interface speed per lane (Gbps) D-PHY (Gsp/s) ComboPHY*†	Max System Interface Frequency (MHz)		
									APB (MHz)	DBI (MHz)	DPI/eDPI (MHz)
Combo D-PHY/C-PHY	D-PHY 4 data lanes / C-PHY 3 data lanes	EDPI	-	720p	-	49	15,50	2,5	500	NA	200
			Single DSC	720p	-	328	18,20	1,5	500	NA	106
				4K	-	332	18,40	1,5	500	NA	108
			Dual DSC	720p	-	577	19,60	1,5	500	NA	408
				4K	-	580	19,70	1,5	500	NA	398
			Dual DSC, Dual Port	720p	-	582	29,30	1,5	500	NA	106
				4K	-	579	29,40	1,5	500	NA	105

* Interface speed is limited according to D-PHY v1.2 and C-PHY v1.0 specifications.
† Tests were performed using a 500Mbps/Msp/s step.
** Assuming that 7 compressed lines are stored in the Rate Buffer.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

B

DBI Color Code Mapping Waveforms

This appendix contains the diagrams that shows how the pixel-to-byte conversion is done for each color code mapping supported by the DBI. The diagrams represent the cases where the interface works as Type A with a Fixed E, Type A with clocked E, and Type B.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

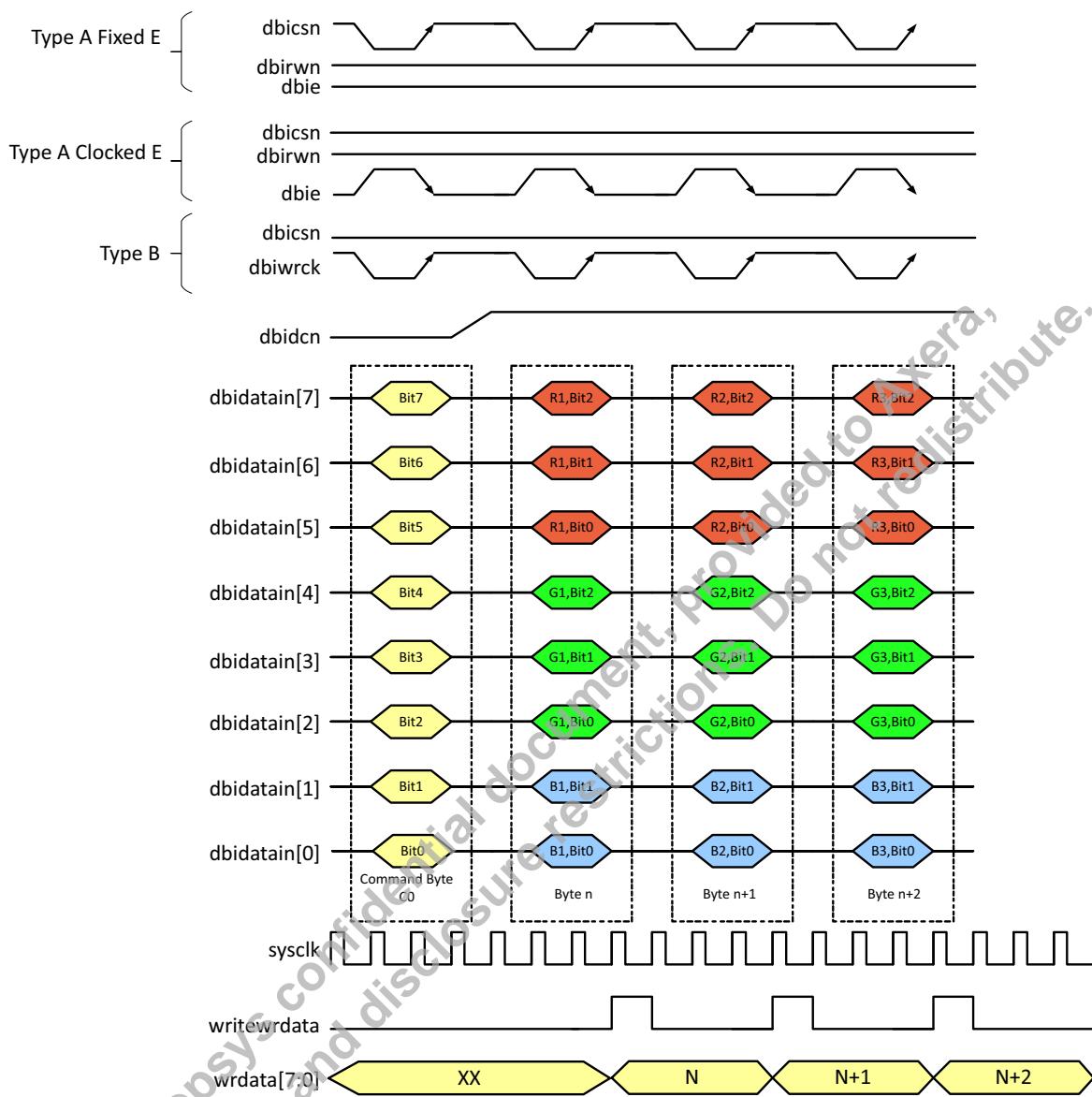
Figure B-1 DSI 8-bit/8-bpp Byte Write

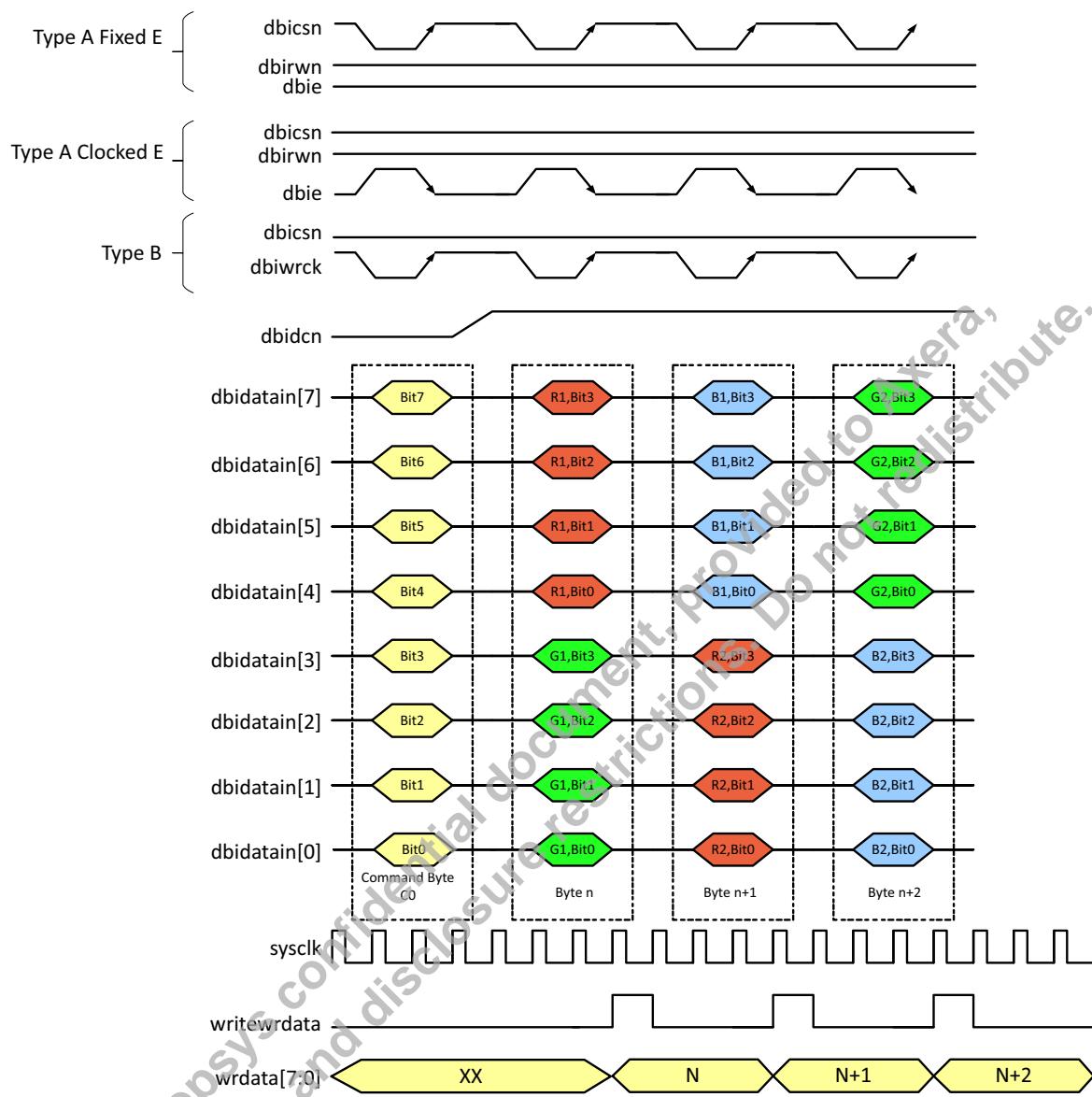
Figure B-2 DSI 8-bit/12-bpp Byte Write

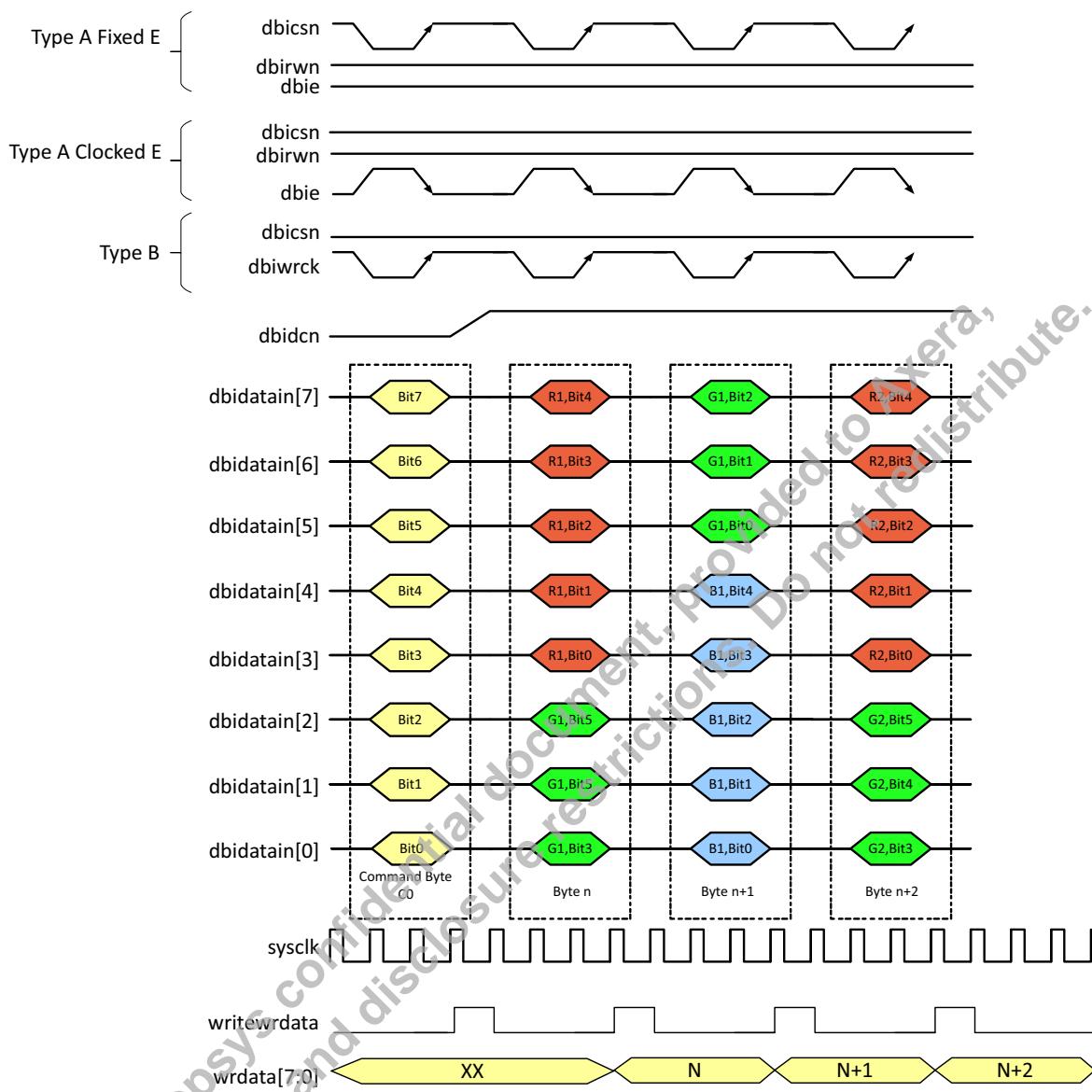
Figure B-3 DSI 8-bit/16-bpp Byte Write

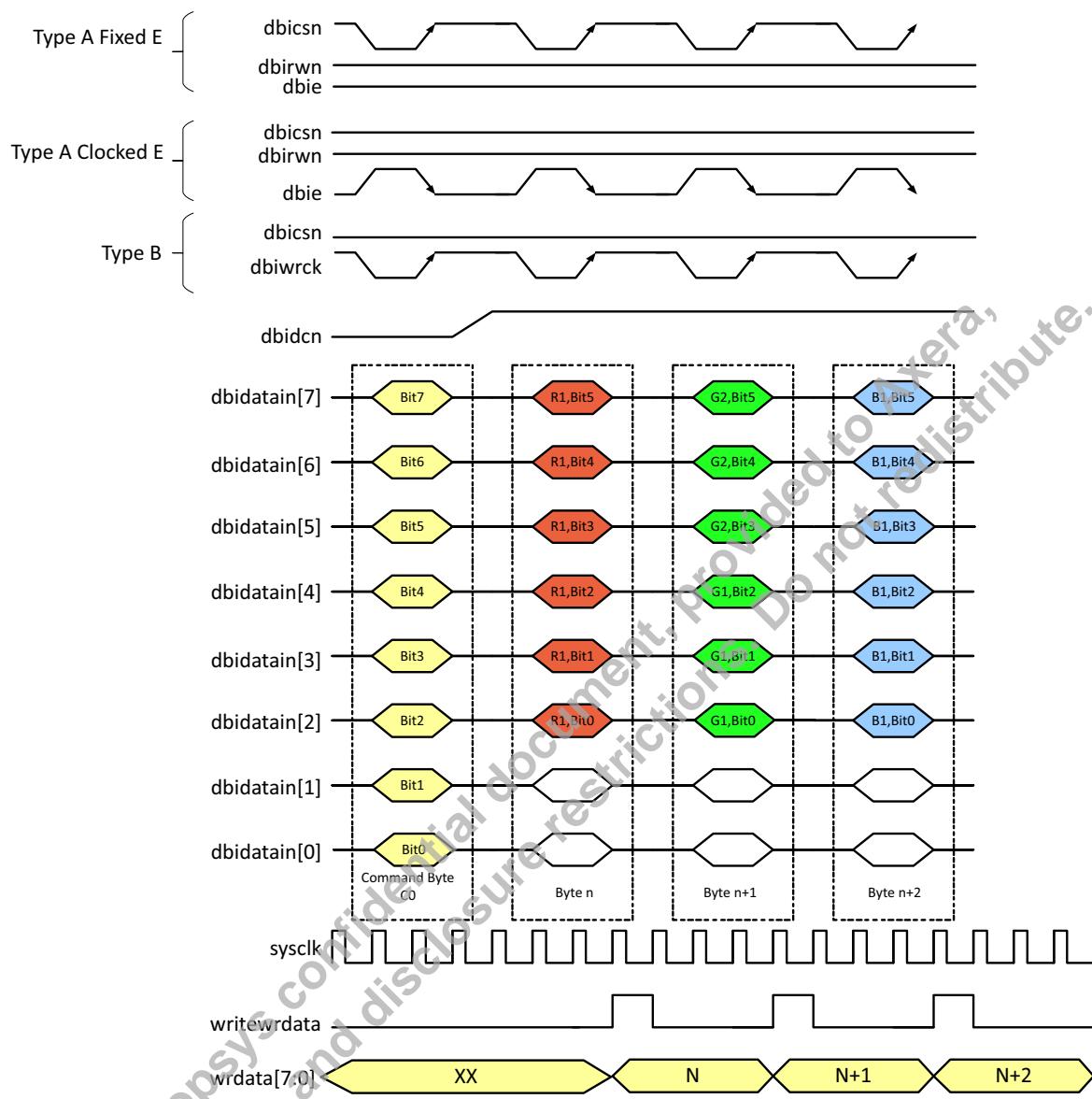
Figure B-4 DSI 8-bit/18-bpp Byte Write

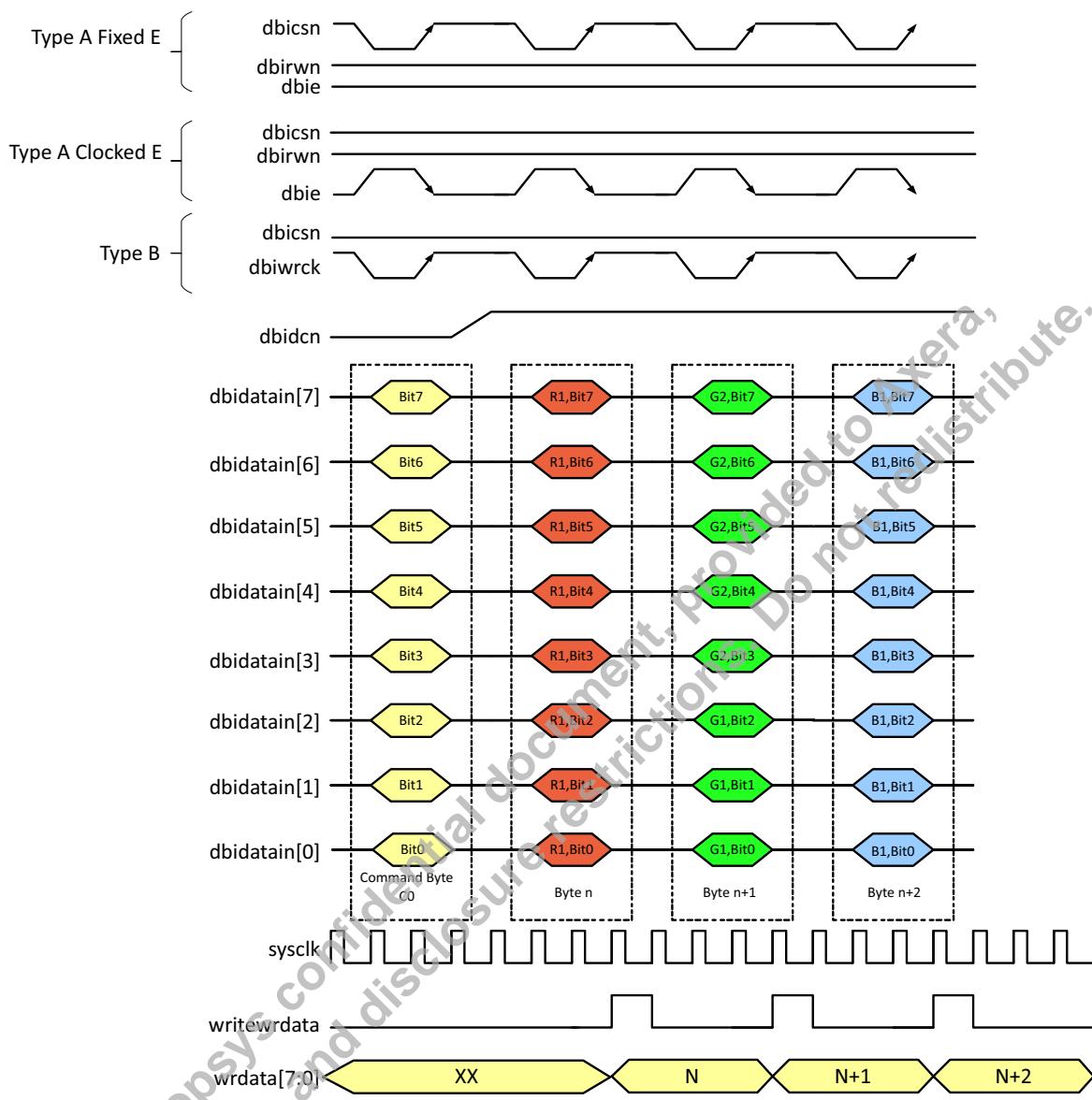
Figure B-5 DSI 8-bit/24-bpp Byte Write

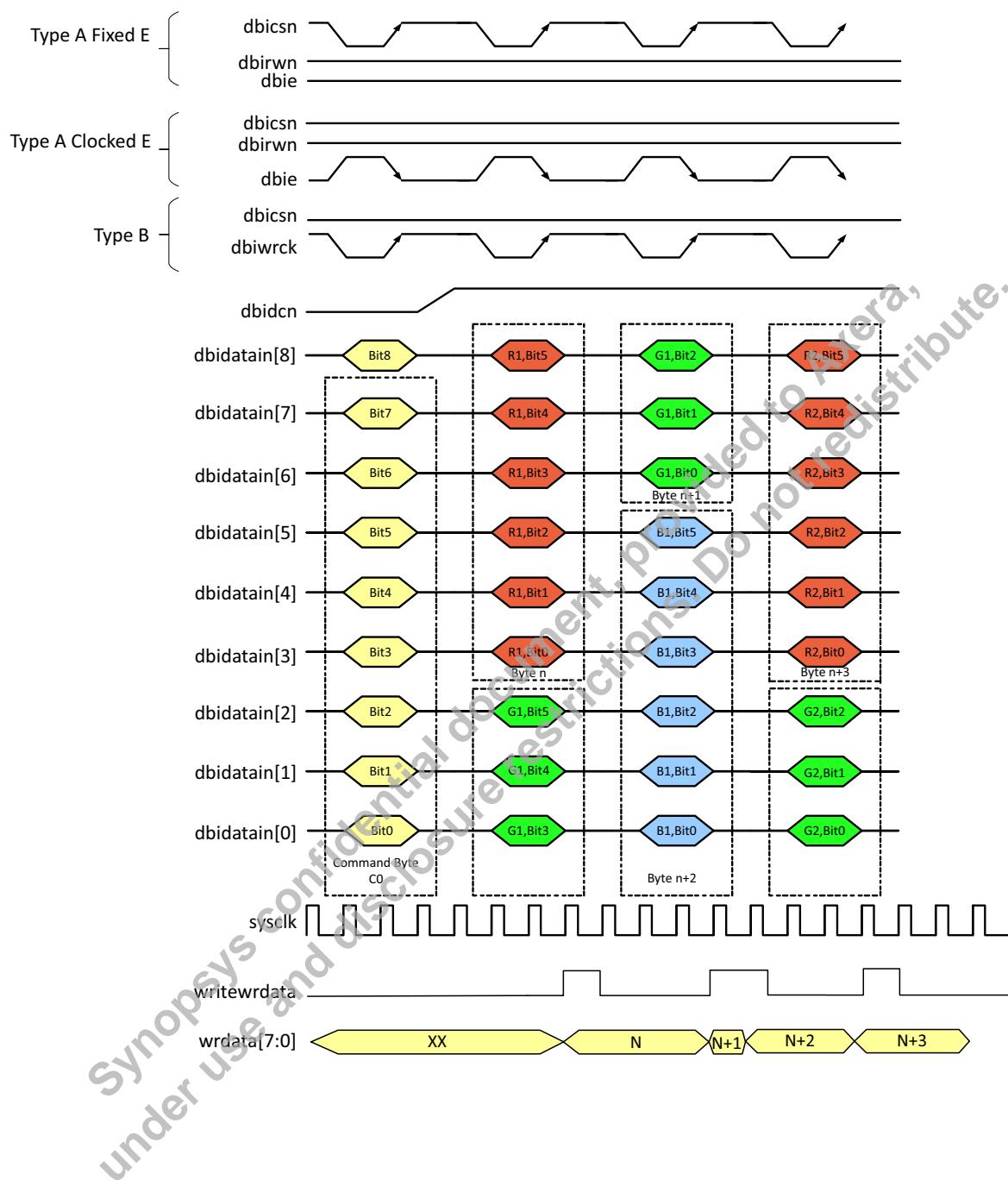
Figure B-6 DSI 9-bit/18-bpp Byte Write

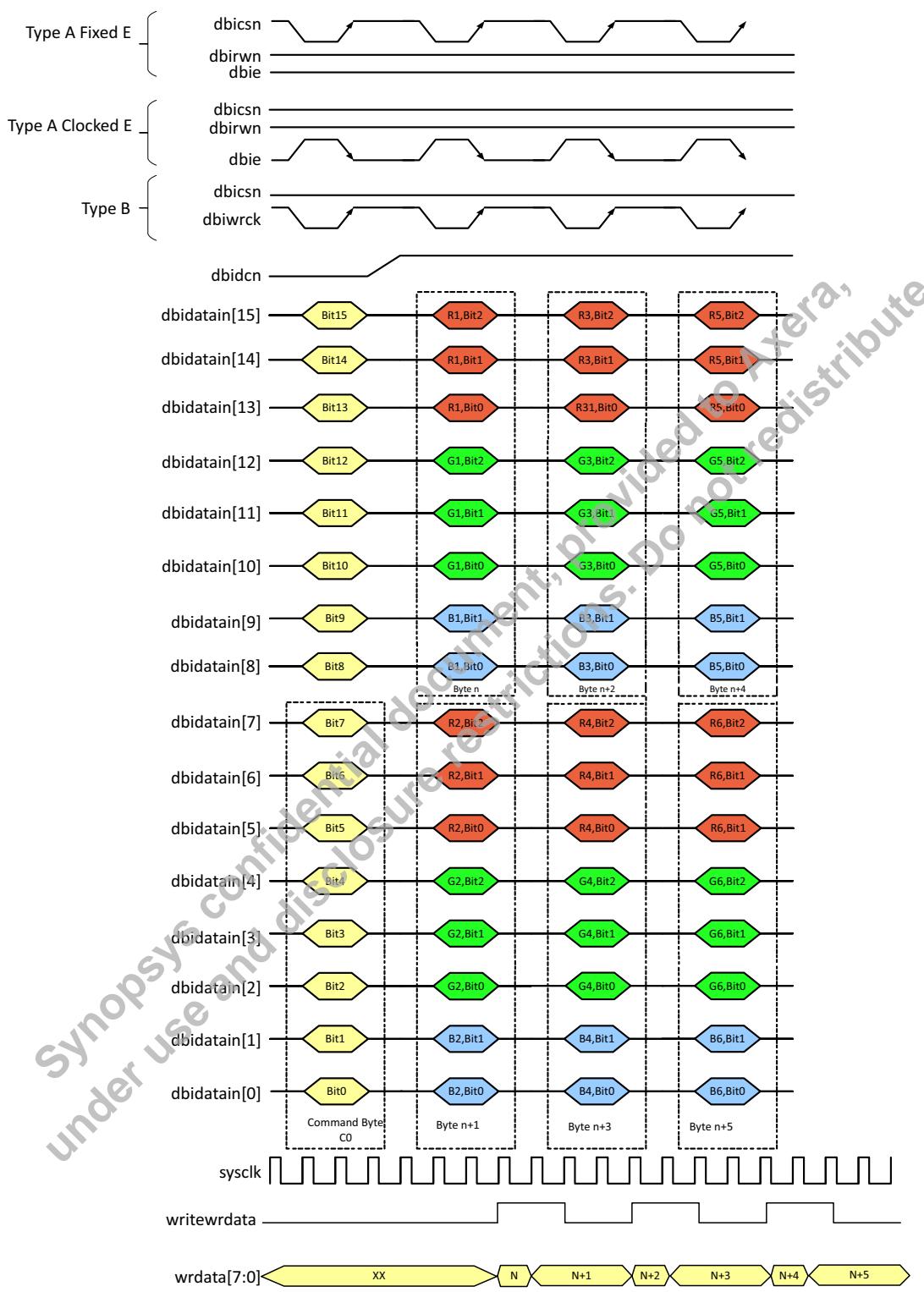
Figure B-7 DSI 16-bit/8-bpp Byte Write

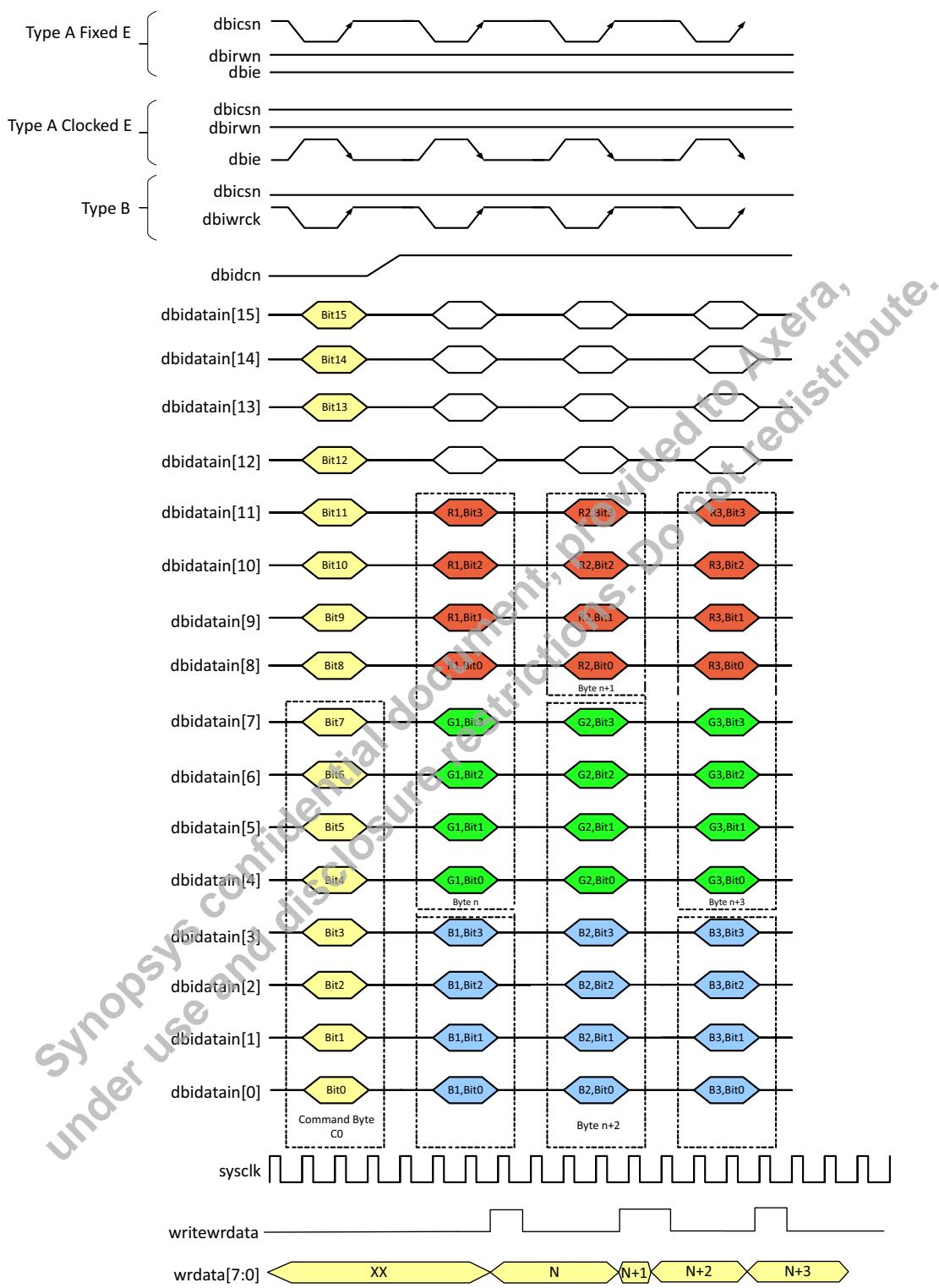
Figure B-8 DSI 16-bit/12-bpp Byte Write

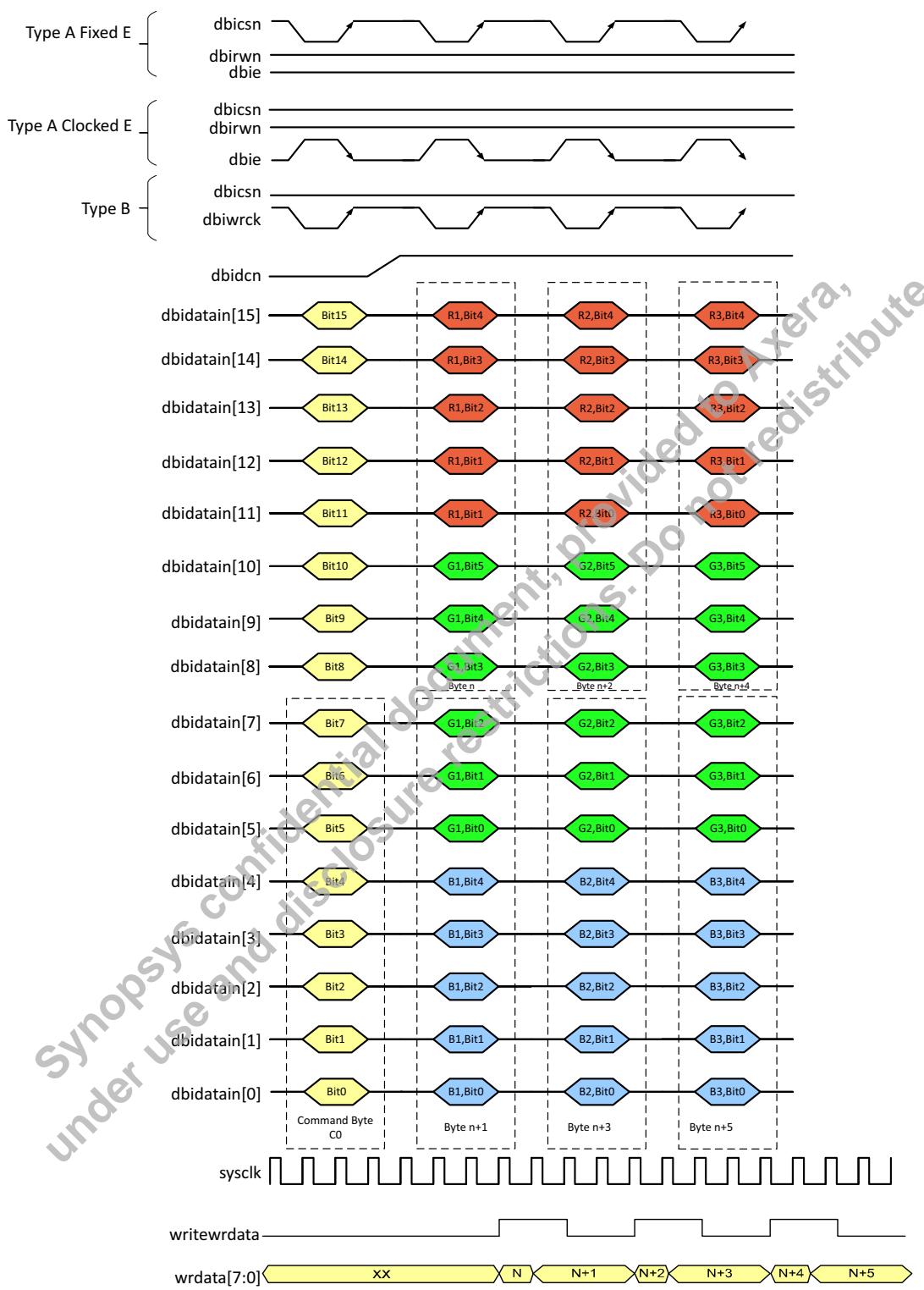
Figure B-9 DSI 16-bit/16-bpp Byte Write

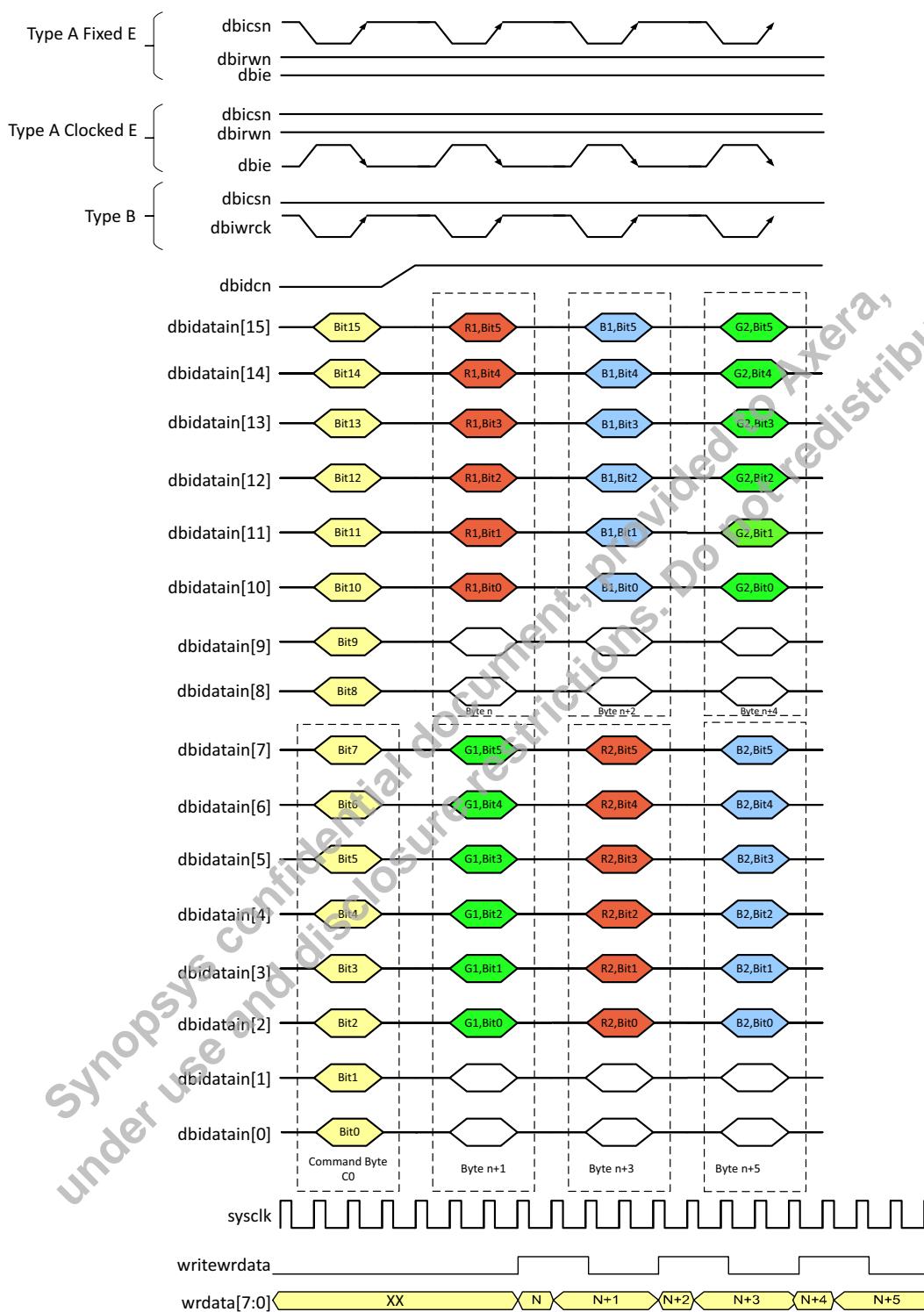
Figure B-10 DSI 16-bit/18-bpp Option 1 Byte Write

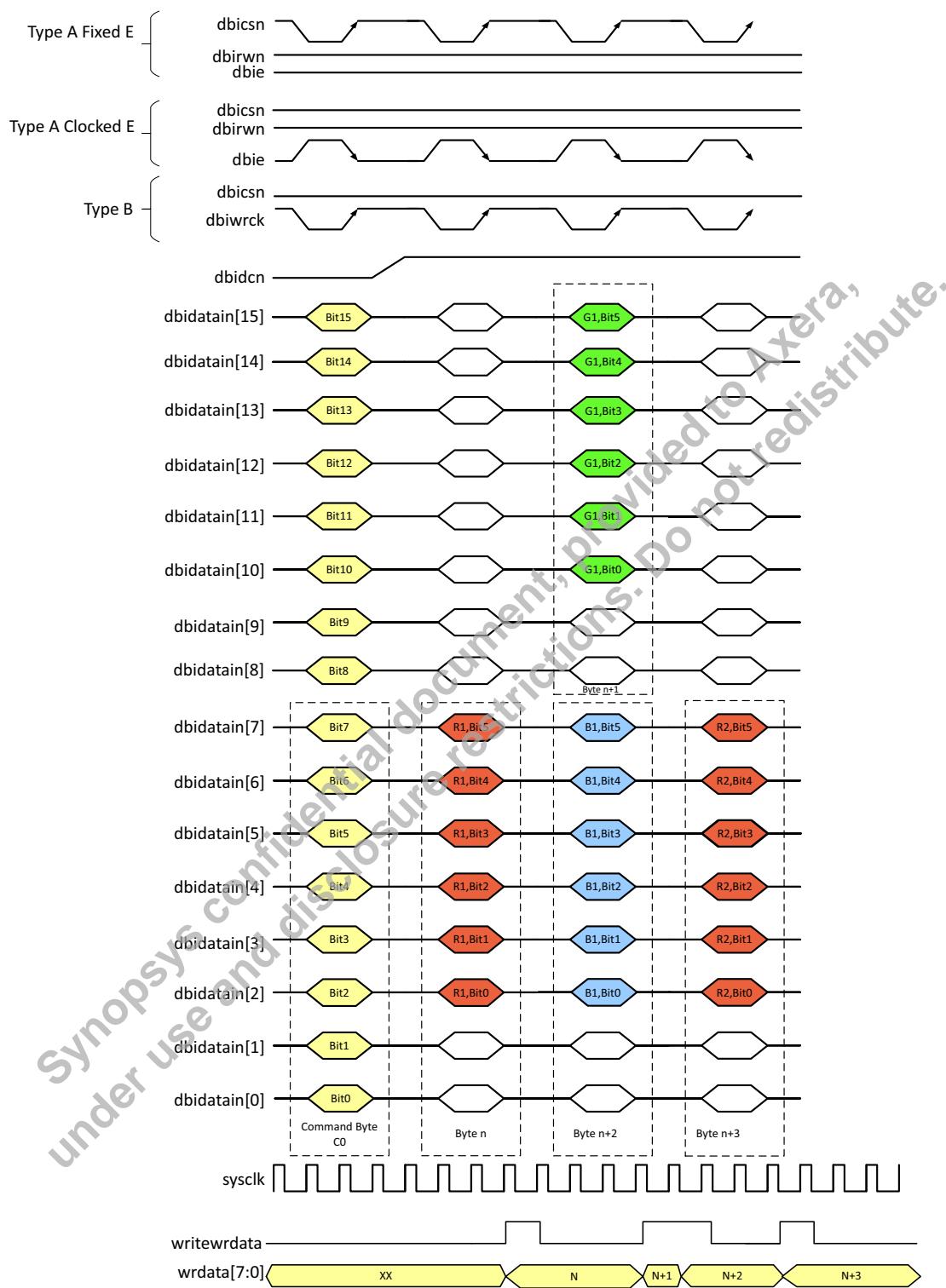
Figure B-11 DSI 16-bit/18-bpp Option 2 Byte Write

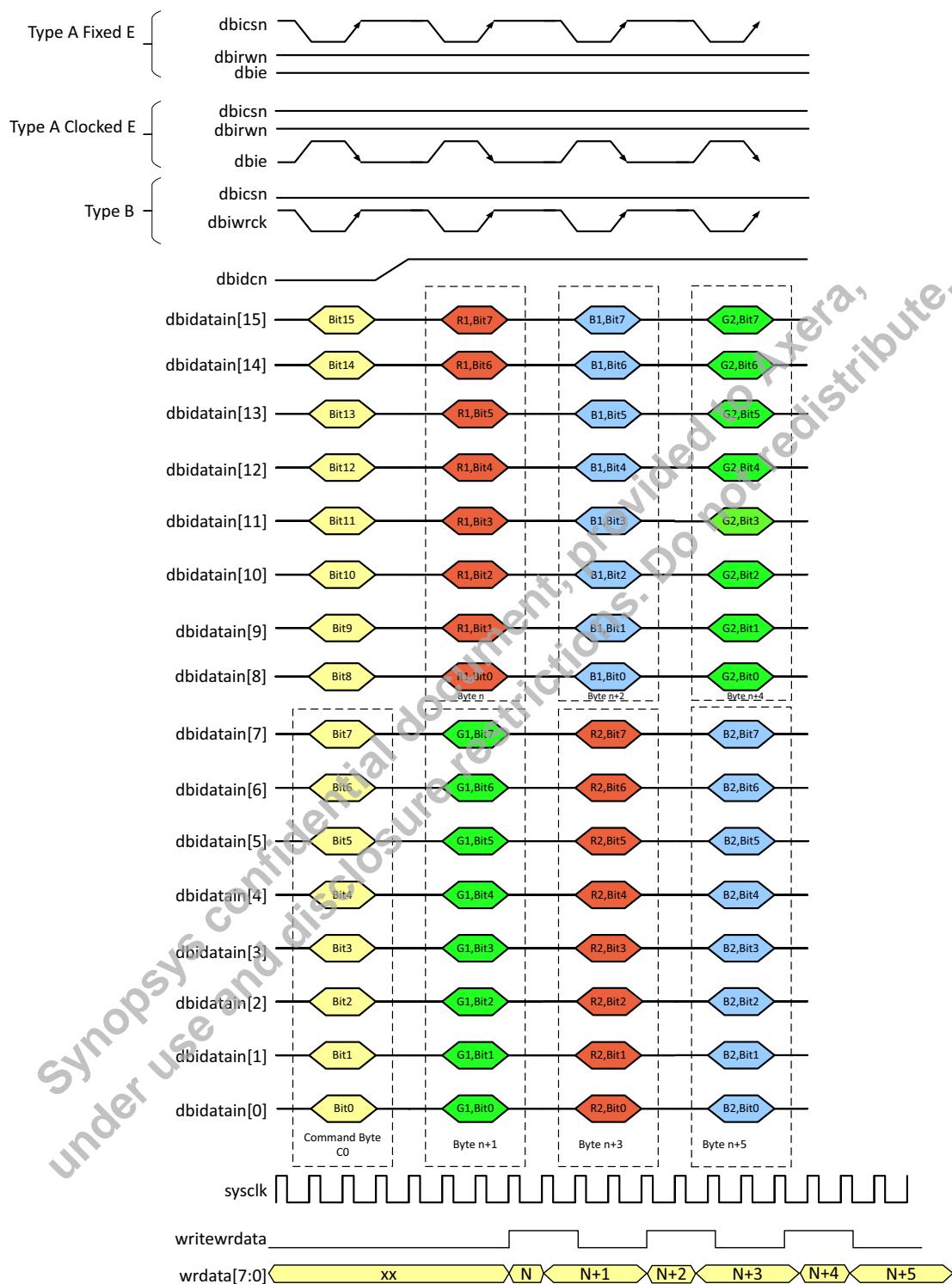
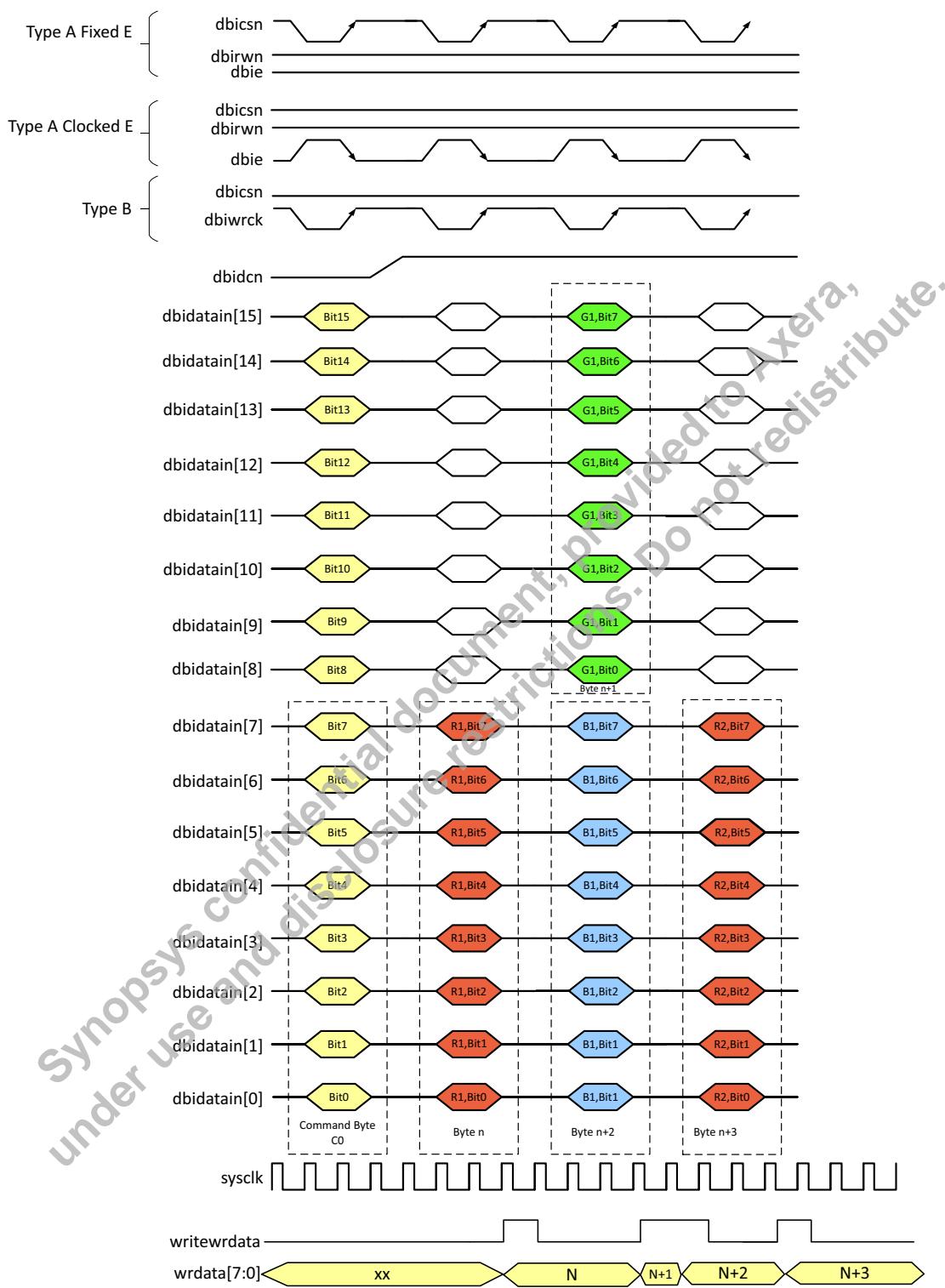
Figure B-12 DSI 16-bit/24-bpp Option 1 Byte Write

Figure B-13 DSI 16-bit/24-bpp Option 2 Byte Write

C

Error Handling

This appendix explains the conditions that trigger the interrupts and provides suggestions to recover from these error states.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

C.1 Causes of Errors and Recovery

The interrupts triggered by the INT_ST0 or INT_ST1 register bits are error conditions that can occur only at the debug stage. “[Error Handling](#)” on page 429 explains the reasons that set off these interrupts and also explains how to recover from these interrupts.



- Though most of the interrupts are only expected to occur during the debug process and should be analyzed and corrected during the design stage, some of the interrupts may occur if the link is exposed to external factors that induce noise in the link.
- If you include VESA DSC encoder, more interrupt sources related to the encoding process exist. These are triggered by the DSC_ENC_INT_ST register.
- In addition to the causes and suggestions for recovery provided in [Table C-1](#) there can be other causes and methods of handling each interrupt.

Table C-1 Error Cause and Recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
INT_ST0 Register Bits			
20	dphy_errors_4	The D-PHY reports the LP1 contention error. The D-PHY host detects the contention while trying to drive the line high.	Recover the D-PHYS from contention. Reset the DWC_mipi_dsi_host controller and transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
19	dphy_errors_3	D-PHY reports the LP0 contention error. The D-PHY Host detects the contention while trying to drive the line low.	Recover the D-PHYS from contention. Reset the DWC_mipi_dsi_host controller and transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
18	dphy_errors_2	The D-PHY reports the False Control Error. The D-PHY detects an incorrect line state sequence in lane 0 lines.	Device does not behave as expected. Communication with the Device is not properly established. This is an unrecoverable error. Reset the DWC_mipi_dsi_host controller and the D-PHY. If this error is recurrent, analyze the behavior of the Device.
17	dphy_errors_1	The D-PHY reports the LPDT Error. The D-PHY detects that the LDPT did not match a multiple of 8 bits.	The data reception is not reliable. The D-PHY recovers but the received data from the Device might not be reliable. It is recommended to reset the DWC_mipi_dsi_host controller and repeat the RX transmission.
16	dphy_errors_0	The D-PHY reports the Escape Entry Error. The D-PHY does not recognize the received Escape Entry Code.	The D-PHY Host does not recognize the Escape Entry Code. The Transmission is ignored. The PHY Host recovers but the system should repeat the RX reception.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
15	ack_with_err_15	This error is directly retrieved from Acknowledge with Error packet. The Device detected a protocol violation in the reception.	See the display documentation. When this error is active, the Device should have another read-back command that reports additional information about this error. Read the additional information and take appropriate actions.
14	ack_with_err_14	The Acknowledge with Error packet contains this error. The Device chooses to use this bit for error report.	See the Device documentation regarding possible reasons for this error and take appropriate actions.
13	ack_with_err_13	The Acknowledge with Error packet contains this error. The Device reports that the transmission length does not match the packet length.	Possible reason for this error is the presence of multiple errors in the packet header (more than 2), therefore, the error detection fails and the Device does not discard the packet. In this case, the packet header is corrupt and can cause decoding mismatches. Transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
12	ack_with_err_12	The Acknowledge with Error packet contains this error. The Device does not recognize the VC ID in at least one of the received packets.	Check the Device capabilities and configure the Host to properly address the Device VC ID. Repeat the transmission.
11	ack_with_err_11	The Acknowledge with Error packet contains this error. The Device does not recognize the data type of at least one of the received packets.	Check the Device capabilities. It is possible that there are some packets that are not supported by the Device. Repeat the transmission.
10	ack_with_err_10	The Acknowledge with Error packet contains this error. The Device detects the CRC errors in at least one of the received packets.	Some of the long packets, transmitted after the last Acknowledge request, might contain the CRC errors in the payload. If the payload content is critical, transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
9	ack_with_err_9	The Acknowledge with Error packet contains this error. The Device detects multi-bit ECC errors in at least one of the received packets.	The Device does not interpret the packets transmitted after the last Acknowledge request. If the packets are critical, transmit the packets again. If this error is recurrent, analyze the connectivity between the Host and the Device carefully.
8	ack_with_err_8	The Acknowledge with Error packet contains this error. The Device detects and corrects the 1 bit ECC error in at least one of the received packets.	No action is required. The Device acknowledges the packet. If this error is recurrent, analyze the signal integrity or the noise conditions of the link.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
7	ack_with_err_7	The Acknowledge with Error packet contains this error. The Device detects the Line Contention through LP0/LP1 detection.	This error might corrupt the low-power data reception and transmission. Ignore the packets and transmit them again. The Device recovers automatically. If this error is recurrent, check the Device capabilities and the connectivity between the Host and Device. See section 7.2.1 of the DSI Specification 1.1.
6	ack_with_err_6	The Acknowledge with Error packet contains this error. The Device detects the False Control Error.	<p>The device detects one of the following:</p> <ul style="list-style-type: none"> ■ The low-power-10 (low-power request) is not followed by the remainder of a valid escape or turnaround sequence. ■ The low-power-01 (high-speed request) is not followed by a bridge state (low-power-00). <p>The PHY communications are corrupted. This error is unrecoverable. Reset the DWC_mipi_dsi_host controller and the PHY. See section 7.1.6 of the DSI Specification 1.1.</p>
5	ack_with_err_5	The Acknowledge with Error packet contains this error. The display timeout counters for a high-speed reception and low-power transmission expire.	<p>It is possible that the Host and Device timeout counters are not correctly configured. The Device HS_TX timeout should be shorter than the Host HS_RX timeout. Host LP_RX timeout should be longer than the Device LP_TX timeout.</p> <p>Check and confirm that the Host configuration is consistent with the Device specifications. This error is automatically recovered, although there is no guarantee that all the packets in the transmission or reception are complete. For additional information about this error, see section 7.2.2 of the DSI Specification 1.1.</p>
4	ack_with_err_4	The Acknowledge with Error packet contains this error. The Device reports that the LPDT is not aligned in an 8-bit boundary.	There is no guarantee that the Device properly receives the packets. Transmit the packets again. For additional information about this error, see section 7.1.5 of the DSI Specification.
3	ack_with_err_3	The Acknowledge with Error packet contains this error. The Device does not recognize the Escape Mode Entry command.	The Device does not recognize the Escape Mode Entry code. Check the Device capability. For additional information about this error, see section 7.1.4 of the DSI Specification. Repeat the transmission to the Device.
2	ack_with_err_2	The Acknowledge with Error packet contains this error. The Device detects the high-speed transmission did not end in an 8-bit boundary when the EoT sequence is detected.	There is no guarantee that the Device properly received the packets. Re-transmission should be performed. Transmit the packets again. For additional information about this error, see section 7.1.3 of the DSI Specification 1.1.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
1	ack_with_err_1	The Acknowledge with Error packet contains this error. The Device detects that the SoT leader sequence is corrupted.	The Device discards the incoming transmission. Re-transmission should be performed by the Host. For additional information about this error, see section 7.1.2 of the DSI Specification 1.1.
0	ack_with_err_0	The Acknowledge with Error packet contains this error. The Device reports that the SoT sequence is received with errors but synchronization can still be achieved.	The Device is tolerant to single bit and some multi-bit errors in the SoT sequence. Yet, the packet correctness is compromised. If the packet content is important, transmit the packets again. For additional information about this error, see section 7.1.1 of the DSI Specification 1.1.
INT_ST1 Register Bits			
17	dbi_illegal_comm_err	A command that is not defined in the DCS Specification is driven in the DBI.	The system should check the DBI usage procedures. Commands that are not defined in the DCS Specification cannot be driven through the DBI.
16	dbi_pld_recv_err	An underflow occurs in the DBI read FIFO.	Read FIFO size is not correctly dimensioned for the maximum read-back packet size. Configure the Device to return the read data with a suitable size for the Host dimensioned FIFO. Data is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the read procedure.
15	dbi_pld_rd_err	An underflow occurs in the DBI read FIFO.	System does not wait for the read procedure to end and starts retrieving the data from the FIFO. The read data is requested before it is received fully in the Host. Data is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the read procedure. Ensure that the read procedure is completed before reading the data through the DBI.
14	dbi_pld_wr_err	An overflow occurs in the DBI write payload FIFO.	The payload FIFO size is not correctly dimensioned to store the total payload of a burst of write packets. For FIFO dimensioning recommendations, see “Parameter Descriptions” chapter. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the write procedure.
13	dbi_cmd_wr_err	An overflow occurs in the DBI command FIFO.	The command FIFO size is not correctly dimensioned to store the total headers of a burst of write packets. For FIFO dimensioning recommendations. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the write procedure.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
12	gen_pld_recv_err	An overflow occurs in the Generic read FIFO.	The Read FIFO size is not correctly dimensioned for the maximum read-back packet size. Configure the Device to return the read data with a suitable size for the Host dimensioned FIFO. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the read procedure.
11	gen_pld_rd_err	An underflow occurs in the Generic read FIFO.	System does not wait for the read procedure to end and starts retrieving the data from the FIFO. The read data is requested before it is fully received. Data is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the read procedure. Check that the read procedure is completed before reading the data through the APB interface.
10	gen_pld_send_err	An underflow occurs in the Generic write payload FIFO.	The system writes the packet header before the respective packet payload is completely loaded into the payload FIFO. This error is unrecoverable, the transmitted packet is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the write procedure.
9	gen_pld_wr_err	An overflow occurs in the Generic write payload FIFO.	The payload FIFO size is not correctly dimensioned to store the total payload of a long packet. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the write procedure.
8	gen_cmd_wr_err	An overflow occurs in the Generic command FIFO.	The command FIFO size is not correctly dimensioned to store the total headers of a burst of packets. For FIFO dimensioning recommendations. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the write procedure.
7	dpi_pld_wr_err	An overflow occurs in the DPI pixel payload FIFO.	The controller FIFO dimensions are not correctly set up for the operating resolution. Check the Video Mode configuration registers. They should be consistent with the DPI video resolution. The pixel data sequence is corrupted. Reset the DWC_mipi_dsi_host controller and re-initiate the Video transmission.
6	eopt_err	Host receives a transmission that does not end with an End of Transmission packet.	This error is not critical for the data integrity of the received packets. Check if the Device supports the transmission of EoTp packets.
5	pkt_size_err	Host receives a transmission that does not end in the expected byte boundaries.	The integrity of the received data cannot be guaranteed. Reset the DWC_mipi_dsi_host controller and repeat the read procedure.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
4	crc_err	Host reports that a received long packet has a CRC error in its payload.	The received payload data is corrupted. Reset the DWC_mipi_dsi_host controller and repeat the read procedure. If this error is recurrent, check the DSI connectivity link for the noise levels.
3	ecc_multi_err	Host reports that a received packet contains multiple ECC errors.	The received packet is corrupted. The DWC_mipi_dsi_host controller ignores all the following packets. The DWC_mipi_dsi_host controller must repeat the read procedure.
2	ecc_single_err	Host reports that a received packet contains a single bit error.	This error is not critical because the DWC_mipi_dsi_host controller can correct the error and properly decode the packet. If this error is recurrent, check the DSI connectivity link for signal integrity and noise levels.
1	to_lp_rx	Host reports that the configured timeout counter for the low-power reception has expired.	Once the configured timeout counter ends, the DWC_mipi_dsi_host controller automatically resets and recovers to normal operation. Packet transmissions happening during this event are lost. If this error is recurrent, check the timer configuration for any issue. This timer should be greater than the maximum low-power transmission generated by the Device.
0	to_hs_tx	Host reports that the configured timeout counter for the high-speed transmission has expired.	Once the configured timeout counter ends, the DWC_mipi_dsi_host controller automatically resets and recovers to normal operation. Packet transmissions happening during this event are lost. If this error is recurrent, check the timer configuration for any issue. This timer should be greater than the maximum high-speed transmission bursts generated by the Host.

DSC_ENC_INT_ST_Register Bits

19	Underflow in the demux RAM for DSC Encoder 1	Pixel data is requested by DSC encoder but is not available. Invalid Picture Parameter Settings settings (pic_width / slice_width).	Program Picture Parameter Settings appropriately and restart the controller.
18	Overflow in the demux RAM	AM for DSC Encoder 1 Pixel data is lost in the slice demultiplexer RAM. This RAM distributes pixel data to each DSC encoder instance, so, the bitstream produced is invalid. Invalid Picture Parameter Settings (pic_width / slice_width) can trigger this error.	Program Picture Parameter Settings appropriately and restart the controller.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
17	Underflow in the demux RAM for DSC Encoder 0	Pixel data is requested by DSC encoder but is not available. Invalid Picture Parameter Settings settings (pic_width / slice_width).	Program Picture Parameter Settings appropriately and restart the controller
16	Overflow in the demux RAM for DSC Encoder 0	Pixel data is lost in the slice demultiplexer RAM. This RAM distributes pixel data to each DSC encoder instance, so, the bitstream produced is invalid. Invalid Picture Parameter Settings (pic_width / slice_width) can trigger this error.	Program Picture Parameter Settings appropriately and restart the controller.
15	Overflow in the event delay buffer	During the initial DSC delay period, line events like rising and falling edges of the hsync, vsync, dataen, colorm and shutdn signals are stored in a buffer. Invalid Picture Parameter Settings settings (pic_width, slice_width, pic_height and slice_height) and/or invalid DSC Delay value can trigger this error.	Program the Picture Parameter Settings and DSC Delay value appropriately and restart the controller.
14	Line Buffer RAM underflow	Data is lost because new data from line buffer is requested when it is empty.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
13	Line Buffer RAM overflow	Data is lost because new data should be stored to the line buffer when it is full.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
12	Balance FIFO for channel Cg underflow	Data is lost because new data from balance FIFO for Cg channel is requested when it is empty.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
11	Balance FIFO for channel Cg overflow	Data is lost because new data needed to be stored to balance FIFO for Cg channel when it is full.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
10	Balance FIFO for channel Co underflow	Data is lost because new data from balance FIFO for Co channel is requested when it is empty.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
9	Balance FIFO for channel Co overflow	Data is lost because new data needs to be stored to balance FIFO for Co channel when it is full.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
8	Balance FIFO for channel Y underflow	Data is lost because new data from balance FIFO for Y channel is requested when it is empty.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
7	Balance FIFO for channel Y overflow	Data is lost because new data should be stored to balance FIFO for Y channel when it is full.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
6	Syntax Element FIFO underflow	Data is lost because new data from syntax element sizes FIFO is requested when it is empty.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
5	Syntax Element FIFO overflow	Data is lost because new data needs to be stored to the syntax element sizes FIFO when it is full.	Compressed stream is ruined. Program the Picture Parameter Set appropriately and restart the controller.
4	Rate Buffer FIFO underflow	Data is lost because new data from rate buffer is requested when it is empty.	Compressed stream is ruined. Program the Picture Parameter Set and DSC_ENC_DELAY register appropriately and restart the controller.
3	Rate Buffer FIFO overflow	Data is lost because new data should be stored to the rate buffer when it is full.	Compressed stream is ruined. Program the Picture Parameter Set and DSC_ENC_DELAY register appropriately and restart the controller.
2	Slice length does not match PPS settings	The size of an entire frame, seen at DPI, is not a multiple of the size of a slice.	Image content is expected to start shifting spatially. Program the Picture Parameter Set to match the current video configuration, particularly, the slice and picture size.
1	Buffer model underflow	The idealized buffer model has hit the underflow condition. This is probably caused by an excessively high programmed constant bit rate.	Although it might still work, produced bit stream is not guaranteed to be compliant with VESA DSC specification for the current setup. Program a lower value for bits per pixel and double check other parameters of the Picture Parameter Set.
0	RC Buffer Fullness overflow	The Rate Control model has hit the overflow condition, that is, rc_buf_thresh[0...13] and rc_range_parameters[0...14] are not appropriately set, to prevent this situation.	Bitstream is ruined because it is not likely to meet the target size. Reprogram the Picture Parameter set, especially adjusting min_QP and max_QP for the top-most ranges of long-term Rate Control, to quantize more, or decrease their respective rc_buf_thresh value, to make them act earlier. Restart the controller.

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.

D

Static and Dynamic Registers

DWC_mipi_dsi_host controller has two kinds of registers:

- **Static Registers:** Static registers can be configured only when the DWC_mipi_dsi_host controller is in reset state.
- **Dynamic Registers:** Dynamic registers can be configured when needed and do not require the DWC_mipi_dsi_host controller to be idle or in reset state.

The Tables [Table D-1](#) on page 439 through [Table D-5](#) on page 442 lists the static and dynamic registers.

Table D-1 General Registers

Register	Address	Type
VERSION	0x0	Static
PWR_UP	0x4	Static
CLKMGR_CFG	0x8	Static
PCKHDL_CFG	0x2C	Static
MODE_CFG	0x34	Dynamic
CMD_MODE_CFG	0x68	Static
CMD_PKT_STATUS	0x74	Dynamic
TO_CNT_CFG	0x78	Static
BTA_TO_CNT	0x8C	Static
LPCLK_CTRL	0x94	Static
DSC_PARAMETER	0xF0	Static
HS_RD_TO_CNT	0x7C	Static
HS_WR_TO_CNT	0x84	Static
LP_RD_TO_CNT	0x80	Static

Register	Address	Type
LP_WR_TO_CNT	0x88	Static

Table D-2 lists the registers related to PHY.

Table D-2 Register Related to PHY

Register	Address	Type
PHY_RSTZ	0xA0	Dynamic
PHY_IF_CFG	0xA4	Static
PHY_ULPS_CTRL	0xA8	Dynamic
PHY_TX_TRIGGER	0xAC	Dynamic
PHY_STATUS	0xB0	Dynamic
PHY_TST_CTRL0	0xB4	Dynamic
PHY_TST_CTRL1	0xB8	Dynamic
PHY_CAL	0xCC	Dynamic
PHY_MODE	0xFC	Static

Table D-3 lists the registers related to interrupt.

Table D-3 Registers Related to Interrupt

INT_ST0	0xBC	Dynamic
INT_ST1	0xC0	Dynamic
INT_MSK0	0xC4	Dynamic
INT_MSK1	0xC8	Dynamic
INT_FORCE0	0xD8	Dynamic
INT_FORCE1	0xDC	Dynamic

Table D-3 lists the registers related to interfaces.

Table D-4 Registers Related to Interfaces

Register	Address	Type
GEN_VCID	0x30	Static
GEN_HDR	0x6c	Dynamic
GEN_PLD_DATA	0x70	Dynamic
DBI_VCID	0x1c	Static

Register	Address	Type
DBI_CFG	0x20	Static
DBI_PARTITIONING_EN	0x24	Static
DBI_CMDSIZE	0x28	Static
DPI_VCID	0xC	Static
DPI_COLOR_CODING	0x10	Static
DPI_CFG_POL	0x14	Static
DPI_LP_CMD_TIM	0x18	Static
EDPI_CMD_SIZE	0x64	Static
AUTO_ULPS_MODE	0xE0	Static
AUTO_ULPS_ENTRY_DELAY	0xE4	Static
AUTO_ULPS_WAKEUP_TIME	0xE8	Static
EDPI_ADV_FEATURES	0xEC	Static
AUTO_ULPS_MIN_TIME	0xF8	Static
EDPI_TE_HW_CFG	0x11C	Static
VID_MODE_CFG	0x38	Static
VID_PKT_SIZE	0x3C	Static
VID_NUM_CHUNKS	0x40	Static
VID_NULL_SIZE	0x44	Static
VID_HSA_TIME	0x48	Static
VID_HPB_TIME	0x4C	Static
VID_HLINE_TIME	0x50	Static
VID_VSA_LINES	0x54	Static
VID_VBP_LINES	0x58	Static
VID_VFP_LINES	0x5C	Static
VID_VACTIVE_LINES	0x60	Static
SDF_3D	0x90	Dynamic
PHY_TMR_LPCLK_CFG	0x98	Static
PHY_TMR_CFG	0x9C	Static
PHY_TMR_RD_CFG	0xF4	Static

Register	Address	Type
VID_SHADOW_CTRL	0x100	Dynamic
DPI_VCID_ACT	0x10C	Dynamic
DPI_COLOR_CODING_ACT	0x110	Static
DPI_LP_CMD_TIM_ACT	0x118	Static
VID_MODE_CFG_ACT	0x138	Dynamic
VID_PKT_SIZE_ACT	0x13C	Dynamic
VID_NUM_CHUNKS_ACT	0x140	Dynamic
VID_NULL_SIZE_ACT	0x144	Static
VID_HSA_TIME_ACT	0x148	Dynamic
VID_HBP_TIME_ACT	0x14C	Dynamic
VID_HLINE_TIME_ACT	0x150	Dynamic
VID_VSA_LINES_ACT	0x154	Static
VID_VBP_LINES_ACT	0x15C	Dynamic
VID_VFP_LINES_ACT	0x158	Dynamic
VID_VACTIVE_LINES_ACT	0x160	Dynamic
VID_PKT_STATUS	0x168	Dynamic
SDF_3D_ACT	0x190	Dynamic

Table D-5 lists the registers specific to DSC encoder.

Table D-5 Registers specific to DSC Encoder (Continued)

Register	Address	Type
DSC_ENC_COREID	0x200	Static
DSC_ENC_VERSION	0x204	Static
DSC_ENC_FLATNESS_DET_THRESH	0x208	Static
DSC_ENC_DELAY	0x20C	Static
DSC_ENC_COMPRESSED_LINE_SIZE	0x210	Static
DSC_ENC_LINES_IN_EXCESS	0x214	Static
DSC_ENC_RBUF_ADDR_LAST_LINE_ADJ	0x218	Static

Register	Address	Type
DSC_MODE	0x21C	Static
DSC_ENC_INT_ST	0x220	Dynamic
DSC_ENC_INT_MSK	0x224	Dynamic
DSC_ENC_INT_FORCE	0x228	Dynamic
DSC_FIFO_STATUS_SELECT	0x22C	Dynamic
DSC_FIFO_STATUS	0x230	Dynamic
DSC_FIFO_STATUS2	0x234	Dynamic
DSC_ENC_PPS_0_3	0x260	Dynamic
DSC_ENC_PPS_4_7	0x264	Dynamic
DSC_ENC_PPS_8_11	0x268	Dynamic
DSC_ENC_PPS_12_15	0x26C	Dynamic
DSC_ENC_PPS_16_19	0x270	Dynamic
DSC_ENC_PPS_20_23	0x274	Dynamic
DSC_ENC_PPS_24_27	0x278	Dynamic
DSC_ENC_PPS_28_31	0x27C	Dynamic
DSC_ENC_PPS_32_35	0x280	Dynamic
DSC_ENC_PPS_36_39	0x284	Dynamic
DSC_ENC_PPS_40_43	0x288	Dynamic
DSC_ENC_PPS_44_47	0x28C	Dynamic
DSC_ENC_PPS_48_51	0x290	Dynamic
DSC_ENC_PPS_52_55	0x294	Dynamic
DSC_ENC_PPS_56_59	0x298	Dynamic
DSC_ENC_PPS_60_63	0x29C	Dynamic
DSC_ENC_PPS_64_67	0x2A0	Dynamic
DSC_ENC_PPS_68_71	0x2A4	Dynamic
DSC_ENC_PPS_72_75	0x2A8	Dynamic
DSC_ENC_PPS_76_79	0x2AC	Dynamic
DSC_ENC_PPS_80_83	0x2B0	Dynamic
DSC_ENC_PPS_84_87	0xB4	Dynamic

Table D-6 Registers Specific to Automotive Package

Register	Address	Type
INT_ST0_AP	0x300	Dynamic
INT_MSK0_AP	0x304	Dynamic
INT_FORCE0_AP	0x308	Dynamic
INT_ST1_AP	0x310	Dynamic
INT_MSK1_AP	0x314	Dynamic
INT_FORCE1_AP	0x318	Dynamic
INT_ST2_AP	0x320	Dynamic
INT_MSK2_AP	0x324	Dynamic
INT_FORCE2_AP	0x328	Dynamic
TO_HSTXRDY_CFG_AP	0x340	Static
TO_LPTXRDY_CFG_AP	0x344	Static
TO_LPTXTRIG_CFG_AP	0x348	Static
TO_LPTXULPS_CFG_AP	0x34C	Static
TO_HSTX_CFG_AP	0x350	Static
TO_HSTX_CTRL_AP	0x354	Static
TO_BTA_CFG_AP	0x358	Static
TO_CLK_DIV_AP	0x35C	Static
ERR_INJ_CTRL_AP	0x380	Static
ERR_INJ_CHK_MSK_AP	0x384	Static
ERR_INJ_DATA_MSK_AP	0x388	Static
ERR_INJ_ST_AP	0x38c	Static

E

Internal Parameter Descriptions

Provides a description of the internal parameters that might be indirectly referenced in expressions in the Signals, Parameters, or Registers chapters. These parameters are not visible in the coreConsultant GUI and most of them are derived automatically from visible parameters. **You must not set any of these parameters directly.**

Some expressions might refer to TCL functions or procedures (sometimes identified as **function_of**) that coreConsultant uses to make calculations. The exact formula used by these TCL functions is not provided in this chapter. However, when you configure the core in coreConsultant, all TCL functions and parameters are evaluated completely; and the resulting values are displayed where appropriate in the coreConsultant GUI reports.

Table E-1 Internal Parameters

Parameter Name	Equals To
DSC_BALANCE_FIFO_SIZE	[function_of: DSC_MAX_MUX_WORD_SIZE DSC_MAX_SE_SIZE]
DSC_BALANCE_RAM_AWIDTH	[function_of: DSC_BALANCE_RAM_DEPTH]
DSC_BALANCE_RAM_COMP_AWIDTH	[function_of: DSC_NUM_COMPONENTS DSC_BALANCE_RAM_AWIDTH]
DSC_BALANCE_RAM_COMP_DWIDTH	[function_of: DSC_NUM_COMPONENTS DSC_BALANCE_RAM_DWIDTH]
DSC_BALANCE_RAM_DWIDTH	32
DSC_LB_NUM_COMPONENTS	3
DSC_LB_RAM_AWIDTH	[function_of: DSC_LB_RAM_DEPTH]
DSC_LB_RAM_DWIDTH	((DSC_MAX_BPC +1) * DSC_LB_NUM_COMPONENTS)
DSC_MAX_BPC	10
DSC_MAX_MUX_WORD_SIZE	((DSC_MAX_BPC >= 12) ? 64 : 48)
DSC_MAX_SE_SIZE	[function_of: DSC_MAX_BPC]

Parameter Name	Equals To
DSC_NUM_COMPONENTS	3
DSC_PIX_DWIDTH	[function_of: DSC_MAX_BPC * DSC_NUM_COMPONENTS]
DSC_SE_SIZE_FIFO_SIZE	[function_of: DSC_MAX_MUX_WORD_SIZE DSC_MAX_SE_SIZE]
DSC_SE_SIZE_RAM_AWIDTH	[function_of: DSC_SE_SIZE_RAM_DEPTH]
DSC_SE_SIZE_RAM_DWIDTH	18
DSC_SLICE_DEMUX_RAM_AWIDTH	[function_of: DSC_SLICE_DEMUX_RAM_DEPTH]
DSC_SLICE_DEMUX_RAM_DWIDTH	[function_of: DSI_HOST_DSC_ENC_TYPE DSC_PIX_DWIDTH]
DSI_HOST_64_DATA_INT	((SNPS_RSVDPARAM_1==1) (DSI_HOST_DSC_ENC==1) (DSI_HOST_PHY!=1))
DSI_HOST_AP	0
DSI_HOST_COMBO	(DSI_HOST_PHY == 2)
DSI_HOST_DBINTERFACE	((DSI_HOST_DATAINTERFACE == 1) (DSI_HOST_DATAINTERFACE == 3))
DSI_HOST_DBIPLDADDRDEPTH	[function_of: DSI_HOST_DBIPLD_SIZE 32 1024]
DSI_HOST_DBIPLDADDRWIDTH	((DSI_HOST_DBINTERFACE == 1) ? [function_of: DSI_HOST_DBIPLD_RAM_ADDRDEPTH] : 0)
DSI_HOST_DBIPLDDATAWIDTH	((DSI_HOST_PHY != 1) ? 64 : 32)
DSI_HOST_DBIREADPLDADDRDEPTH	[function_of: DSI_HOST_DBIREADPLD_SIZE 8 1024]
DSI_HOST_DBIREADPLDADDRWIDTH	((DSI_HOST_DBINTERFACE == 1) ? [function_of: DSI_HOST_DBIREADPLD_RAM_ADDRDEPTH] : 0)
DSI_HOST_DPIINTERFACE	((DSI_HOST_DATAINTERFACE == 2) (DSI_HOST_DATAINTERFACE == 3) (DSI_HOST_DATAINTERFACE == 4))
DSI_HOST_DPI_PIXEL_RAM_DWIDTH	((DSI_HOST_AP == 0) ? ((DSI_HOST_64_DATA_INT == 1) ? 64 : 32) : 39)
DSI_HOST_DPI_PLD_FIFO_ADDRDEPTH	[function_of: DSI_HOST_DPI_PLD_SIZE DSI_HOST_DPI_RB_RAM_DATAWIDTH 4096]
DSI_HOST_DPI_RB_RAM_ADDRWIDTH	[function_of: DSI_HOST_DPI_RB_RAM_ADDRDEPTH]
DSI_HOST_DPI_RB_RAM_DATAWIDTH	(DSI_HOST_64_DATA_INT == 1) ? 64 : 32
DSI_HOST_DUAL_DSC_ENC	(DSI_HOST_DSC_ENC_TYPE != 1)
DSI_HOST_DUAL_DSC_ENC_DPORT	(DSI_HOST_DSC_ENC_TYPE == 4)

Parameter Name	Equals To
DSI_HOST_DUAL_DSC_ENC_INCDEMUX	(DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX == 1) (DSI_HOST_DUAL_DSC_ENC_SPORT_DPIX == 1)
DSI_HOST_DUAL_DSC_ENC_INC_DSCCLK	(DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX == 1)
DSI_HOST_DUAL_DSC_ENC_SHARE_DEMUX_RAM	(DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX == 1)
DSI_HOST_DUAL_DSC_ENC_SHARE_DPI_RAM	(DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX == 1)
DSI_HOST_DUAL_DSC_ENC_SPORT_DPIX	(DSI_HOST_DSC_ENC_TYPE == 3)
DSI_HOST_DUAL_DSC_ENC_SPORT_SPIX	(DSI_HOST_DSC_ENC_TYPE == 2)
DSI_HOST_EDPIINTERFACE	(DSI_HOST_DATAINTERFACE == 4)
DSI_HOST_GEN31PHY	[function_of:] == 4
DSI_HOST_GEN32PHY	[function_of:] >= 5
DSI_HOST_GENERICPLDADDRDEPTH	[function_of: DSI_HOST_GENERICPLD_SIZE 32 4096]
DSI_HOST_GENERICPLDADDRWIDTH	[function_of: DSI_HOST_GENERICPLD_RAM_ADDRDEPTH]
DSI_HOST_GEN_PLD_RAM_DWIDTH	((DSI_HOST_AP == 0) ? ((DSI_HOST_64_DATA_INT == 1) ? 64 : 32) : 39)
DSI_HOST_GENREADPLDADDRDEPTH	[function_of: DSI_HOST_GENREADPLD_SIZE 32 1024]
DSI_HOST_GENREADPLDADDRWIDTH	[function_of: DSI_HOST_GENREADPLD_RAM_ADDRDEPTH]
DSI_HOST_GEN_READ_RAM_DWIDTH	((DSI_HOST_AP == 1) ? 39 : 32)
DSI_HOST_NUMBER_OF_LANES	(DSI_HOST_DPHY_NUMBER_OF_LANES)
DSI_HOST_PIXELDATAWIDTH	(SNPS_RSVDPARAM_1 == 1) ? 60 : 30
DSI_HOST_PPI_WIDTH	((DSI_HOST_PHY != 1) ? 16 : 8)
DSI_HOST_RB_SEC_DSC_RAM_ADDRWIDTH	[function_of: DSI_HOST_RB_SEC_DSC_RAM_ADDRDEPTH]
DSI_HOST_VERSION_ID	[function_of: -format ASCII -language integer]
SNPS_RSVDPARAM_1	(DSI_HOST_DSC_ENC_TYPE == 3) ? 1 : 0

Synopsys confidential document, provided to Axera,
under use and disclosure restrictions. Do not redistribute.