SYNOPSYS®

DesignWare Cores MIPI DSI Host Controller

Databook

DWC MIPI DSI Host

1.20a
August 2012

Copyright Notice and Proprietary Information

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.

Registered Trademarks (®)

Synopsys, AEON, AMPS, ARC, Astro, Behavior Extracting Synthesis Technology, Cadabra, CATS, Certify, CHIPit, CoMET, Confirma, CODE V, Design Compiler, DesignSphere, DesignWare, Eclypse, Formality, Galaxy Custom Designer, Global Synthesis, HAPS, HapsTrak, HDL Analyst, HSIM, HSPICE, Identify, Leda, LightTools, MAST, MaVeric, METeor, ModelTools, NanoSim, NOVeA, OpenVera, ORA, PathMill, Physical Compiler, PrimeTime, SCOPE, SiVL, SNUG, SolvNet, Sonic Focus, STAR Memory System, SVP Café, Syndicated, Synplicity, the Synplicity logo, Synplify, Synplify Pro, Synthesis Constraints Optimization Environment, TetraMAX, UMRBus, VCS, Vera, and YieldExplorer are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, ASAP, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, BEST, Columbia, Columbia-CE, Cosmos, CosmosLE, CosmosScope, CRITIC, CustomExplorer, CustomSim, Custom WaveView, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, DesignPower, DFTMAX, Direct Silicon Access, Discovery, Encore, EPIC, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, High-performance ASIC Prototyping System, HSIMplus, i-Virtual Stepper, IC Compiler, IICE, in-Sync, iN-Tandem, Intelli, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Macro-PLUS, Magellan, Mars, Mars-Rail, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, MultiPoint, ORAengineering, Physical Analyst, Planet, Planet-PL, Platform Architect, Polaris, Power Compiler, Processor Designer, Raphael, RippledMixer, Saturn, Scirocco, Scirocco-i, SiWare, SPW, Star-RCXT, Star-SimXT, StarRC, Synphony Model System Compiler, System Designer, System Studio, Taurus, TotalRecall, TSUPREM-4, VCSi, VHDL Compiler, VMC, and Worksheet Buffer are trademarks of Synopsys, Inc.

Service Marks (SM)

MAP-in and TAP-in are service marks of Synopsys, Inc.

Third Party Trademark Acknowledgements

SystemC is a trademark of the Open SystemC Initiative and is used under license.

ARM and AMBA are registered trademarks of ARM Limited.

Saber is a registered trademark of SabreMark Limited Partnership and is used under license.

Entrust is a registered trademark of Entrust Inc. in the United States and in certain other countries. In Canada, Entrust is a trademark or registered trademark of Entrust Technologies Limited. Used by Entrust.net Inc. under license.

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

Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Contents

Revision History

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

Date	Version	Description
August 2012	1.20a	Added: ■ A column for the power consumption numbers in Table 1-3. ■ 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" 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. Updated: ■ The "Transmission of Commands in Low-Power" section. ■ The Table 1-2. ■ The size of the dpipixdata[29:0] signal. ■ The descriptions of the presetn, edpite, 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. The reorganization of the register bank impacts the register configuration information in other chapters such as Architecture and Video Interfaces.

Date	Version	Description
April 2012	1.10a	Added: ■ 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 A, “DBI Color Code Mapping Waveforms”. ■ The FIFO depth information to the parameters in Table 2-1. ■ 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 in Table 5-3. ■ The new testcases test_vtb_dpi_generic_lpcmd and test_vtb_edpi_01 in Table 9-2. ■ The eDPI task “write_edpi_packet”. ■ Table 4-2 that shows number of pixels to bytes conversion for the DBI pixels formats. Updated: ■ The “Clock Speed 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 1-3. ■ The “GEN_HDR”, “CLKMGR_CFG”, “PHY_TMR_CFG”, and “GEN_PLD_DATA” registers in the “Register Descriptions” chapter. ■ The DSI_DATAINTERFACE parameter in Table 2-1.

Date	Version	Description
July 2011	1.01a	Added: ■ A new signal phyulpsactivenotclk to Table 5-12. ■ 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. Removed: ■ All references to Asynchronous RAMs as DSI controller now supports both Synchronous and Asynchronous RAMs. Updated: ■ The value after reset of the register Version to 0x3130312A. ■ The area values in Table 1-2.
December 2010	1.00a	Initial release

Preface

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

Databook Organization

The chapters of this databook are organized as follows:

- Chapter 1, "Product Overview", provides an introduction to DWC_mipi_dsi_host, including a block diagram, supported features, deliverables, supported standards, and so on.
- Chapter 2, "Configuration Parameters", describes the hardware configuration parameters.
- Chapter 3, "Architecture", describes the general architecture, startup sequence, and interrupt mechanism of DWC_mipi_dsi_host.
- Chapter 4, "Video Interfaces", describes the video interfaces of DWC_mipi_dsi_host and their functionalities.
- Chapter 5, "Signals", provides descriptions of the inputs/outputs of DWC_mipi_dsi_host.
- Chapter 6, "Register Descriptions", provides the memory map of DWC_mipi_dsi_host and the descriptions of the programmable software registers.
- Appendix A, "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 DBI Interface.
- Appendix B, "Error Handling", explains the conditions that trigger the interrupts and provides suggestions to recover from these error states.

Related Documentation

Refer to the following documentation:

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

Web Resources

The following web links are various Synopsys online resources you may find useful:

- DesignWare IP product information: http://www.designware.com
- Your custom DesignWare IP page: http://www.mydesignware.com

- Documentation through SolvNet: http://solvnet.synopsys.com (Solvnet ID required)
- Synopsys Common Licensing (SCL): http://www.synopsys.com/keys

Customer Support

To obtain support for your product:

- First, prepare the following debug information, if applicable:
  - For environment setup problems or failures with configuration, simulation, or synthesis that occur within coreConsultant or coreAssembler, use the following menu entry:

    File > Build Debug Tar-file

    Check all the boxes in the dialog box that apply to your issue. This menu entry gathers all the Synopsys product data needed to begin debugging an issue and writes it to the file <*core tool startup directory*>/debug.tar.gz.
  - For simulation issues outside of coreConsultant or coreAssembler:
    - Create a waveforms 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
- Then, contact Support Center, with a description of your question and supplying the above information, using one of the following methods:
  - *For fastest response*, use the SolvNet website. If you fill in your information as explained below, your issue is automatically routed to a support engineer who is experienced with your product. The **Sub Product** entry is critical for correct routing.

    Go to http://solvnet.synopsys.com/EnterACall and click on the link to enter a call.
    Provide the requested information, including:
    - **Product**: DesignWare Cores
    - **Sub Product**: MIPI Controller
    - **Tool Version**: 1.20a
    - **Problem Type**:
    - **Priority**:
    - **Title**: dwc_mipi_dsi_host - <Provide a short summary of the issue or list the error message you have encountered>
    - **Description**: For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood

    After creating the case, attach any debug files you created in the previous step.
  - Or, send an e-mail message to support_center@synopsys.com (your e-mail will be queued and then, on a first-come, first-served basis, manually routed to the correct support engineer):
    - Include the Product name, Sub Product name, and Tool Version number in your e-mail (as identified above) 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 in the previous step.

- 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:
    http://www.synopsys.com/Support/GlobalSupportCenters

1 Product Overview

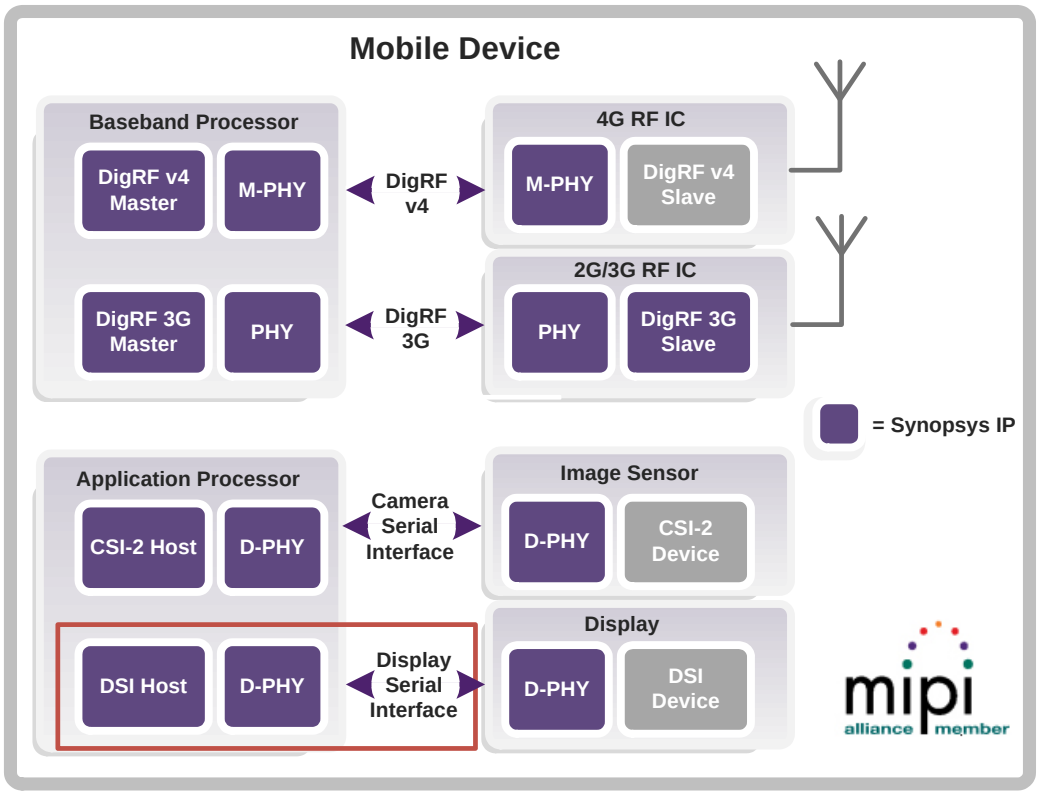
This chapter provides an introduction to the DWC_mipi_dsi_host, including a block diagram, supported features, deliverables, supported standards, and so on. It has the following sections:

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) is a digital core that implements all protocol functions defined in the MIPI DSI Specification. The DWC_mipi_dsi_host provides an interface between the system and the MIPI D-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 in an example system-on-chip design.

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



1.1.1 Applications

Typical applications built with the DWC_mipi_dsi_host are:

- Cellular phones
- Tablets
- Gaming consoles
- PDAs and PMPs
- MIDs

1.1.2 Standards Compliance

The DWC_mipi_dsi_host conforms to the following standards:

- MIPI® Alliance Specification for Display Serial Interface (DSI) v1.1 - 14 March 2012
- MIPI® Alliance Standard for Display Bus Interface v2.00 (DBI-2) - 16 Nov 2005
- MIPI® Alliance Specification for Display Command Set (DCS) v1.1 - 14 March 2012
- MIPI® Alliance Standard for Display Pixel Interface v2.00 (DPI-2) - 23 Jan 2006
- MIPI® Alliance Specification for Stereoscopic Display Formats (SDF) v1.0 - 14 March 2012
- MIPI® Alliance Specification for D-PHY v1.1 - 16 Dec 2011
- AMBA 2.0 Specification (APB) from ARM

1.2 DWC_mipi_dsi_host Features

The DWC_mipi_dsi_host supports the following features:

- Compliant with MIPI Alliance standards (see "Standards Compliance")
- DBI interface 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 twelve. For example, for a maximum D-PHY clock lane speed of 500 MHz, the maximum DBI clock is 41.7 MHz.
- DBI interface 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 interface color coding mappings into 24-bit Interface:
  - 16 bits per pixel, configurations 1, 2, and 3
  - 18 bits per pixel, configurations 1 and 2
  - 24 bits per pixel
- Programmable polarity of all DPI interface signals
- Extended resolutions beyond the DPI standard maximum resolution of 800x480 pixels:
  - Up to 2047 vertical active lines
  - Up to 63 vertical back porch lines

- Up to 63 vertical front porch lines
- Maximum resolution is limited by available DSI Physical link bandwidth:
  - Number of lanes
  - Maximum speed per lane

For a list of supported example resolutions, see "Supported Resolutions and Frame Rates" on page 20.

- Enhanced DPI (eDPI) interface that supports sending large amounts of data through the memory_write_start (WMS) and memory_write_continue (WMC) DCS commands
- Co-existence of DBI and DPI interfaces with only one being operational
- All commands defined in MIPI Alliance Specification for Display Command Set (DCS)
- Interface with MIPI D-PHY following PHY Protocol Interface (PPI), as defined in MIPI Alliance Specification for D-PHY
- Up to four D-PHY Data Lanes
- Bidirectional communication and escape mode support through data lane 0
- Independently programmable virtual channel id associated with the video transmissions through DBI, DPI, eDPI, and APB Slave interfaces
- Transmission of all generic commands
- Transmission of commands in Low-Power in Video mode
- ECC and Checksum capabilities
- End of Transmission Packet (EoTp)
- Ultra Low-Power mode
- Fault recovery schemes

1.2.1 Unsupported Features and Exceptions

The unsupported features and exceptions are as follows:

- 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.
- For the DBI interface, it is necessary to operate the D-PHY clock lane at least twelve times faster than the DBI clock. For example, if the DBI clock is 20 MHz, the D-PHY clock lane must be at least 240 MHz.

1.3 Deliverables

The DWC_mipi_dsi_host is packaged as a .run file. The license file, required to use this Synopsys DesignWare IP, is delivered separately. Use the Synopsys coreConsultant tool to configure, synthesize, and simulate the DWC_mipi_dsi_host core. The DWC_mipi_dsi_host image includes the following:

- Verilog RTL source code
- A Verilog testbench that can be configured to use a generic D-PHY model or a Synopsys Verilog model
- Synthesis scripts for Synopsys Design Compiler and Synplify Pro
- Regression scripts for the following simulators: Synopsys VCS, ModelSim, and NC-Verilog
- Leda checker rules used for linting and example report
- Gate-level validation scripts for:
  - Formal Verification (using Formality)
  - Static timing analyses (using PrimeTime)
  - ATPG patterns verification (using Tetramax)
- DWC_mipi_dsi_host Databook (this document), Installation Guide, and Release Notes

1.4 Operational Model Overview

The DWC_mipi_dsi_host provides means for seamless integration with Synopsys D-PHYs through coreConsultant. Optionally, the DWC_mipi_dsi_host can be configured for a non-Synopsys D-PHY. In such a configuration, it exhibits a PPI-compliant interface to connect to a D-PHY.

The DWC_mipi_dsi_host includes dedicated video interfaces and a generic APB interface that can be used to transmit information to the display. These interfaces are as follows:

- DPI interface: This interface follows the MIPI DPI specification. It is used to transmit information in Video mode, in which the transfers from the host processor to the peripheral take the form of a real-time pixel stream. The DPI interface does not operate concurrently with the DBI interface.
- eDPI interface: This is an enhanced version of the DPI interface, customized for command mode. It is used to transmit information in full bandwidth. It can be used as a fully compliant DPI interface or as Synopsys proprietary interface. It provides support for the standard Video mode and the adapted Command mode and also for the tearing effect.
- DBI interface: This interface follows the MIPI DBI specification. It is used to transmit information in Command mode, where the transactions are commands defined in the DCS specification. The DBI interface does not operate concurrently with DPI interface.
- APB slave interface: This interface allows the transmission of generic information in Command mode, and follows the Synopsys proprietary register interface. This interface can operate concurrently with either DPI or DBI interfaces.

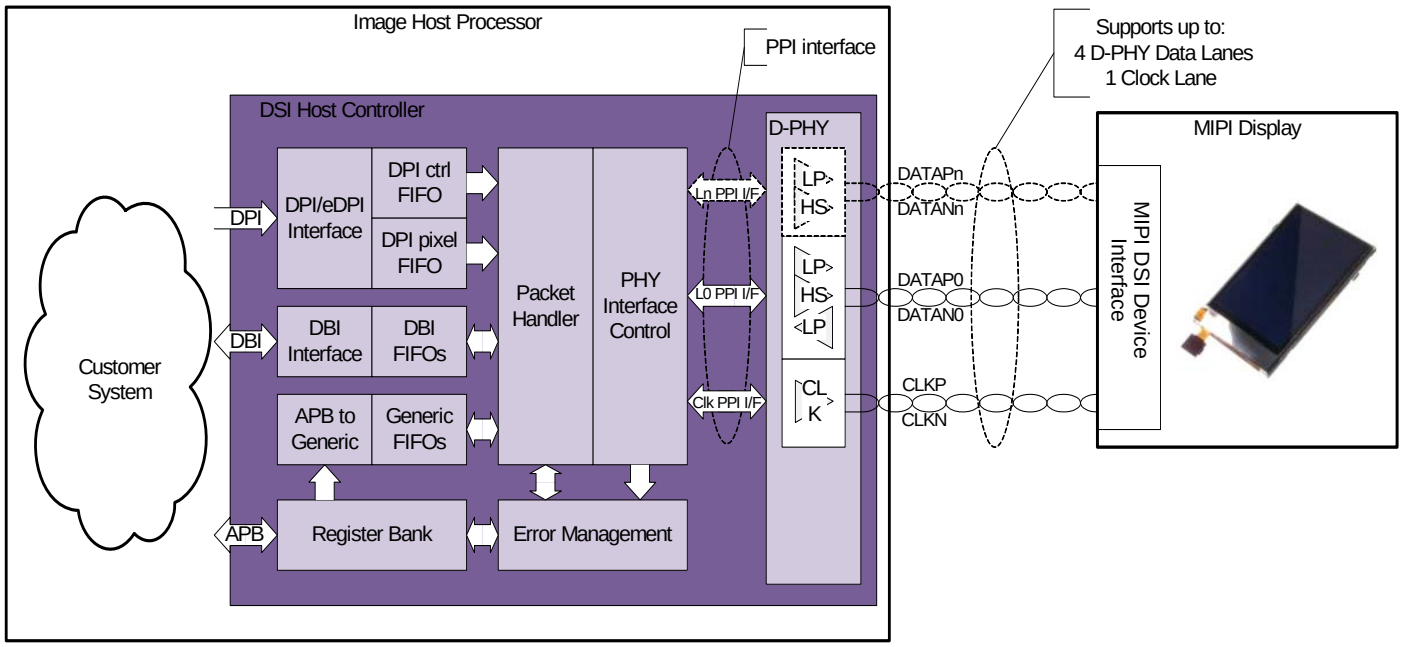
For more information about the DWC_mipi_dsi_host interfaces, see "Video Interfaces" on page 43.

The main functional blocks of DWC_mipi_dsi_host are as follows:

- **D-PHY macro**: Implements the physical layer
- **PHY Interface Control**: Manages the D-PHY PPI interface
- **Packet Handler**: Schedules the activities inside the link
- **Error Management**: Notifies and monitors error conditions on the DSI link
- **APB to Generic**: Bridges the APB operations holding the Command mode commands
- **AMBA-APB Register Bank**: Provides access to the configuration and control registers
- **DPI interface**: Provides a system interface for the video mode operations
- **eDPI interface**: Provides a system interface for the Command mode DCS memory write operations with wider bandwidth using DPI interface signals
- **DBI interface**: Provides a system interface for Command mode with support for DCS commands

Figure 1-2 shows the system-level block diagram of DWC_mipi_dsi_host.

Figure 1-2 Block Diagram of DWC_mipi_dsi_host Functions



1.5 Supported Resolutions and Frame Rates

The DSI specification does not define supported standard resolutions or frame rates. Display resolution, blanking periods, synchronization events duration, frame rates, and pixel color depth play a fundamental role in the required bandwidth. In addition, other link related attributes can influence the ability of the link to support a DSI-specific device. These attributes can be: display input buffering capabilities, video transmission mode (Burst or Non-Burst), Bus Turn-Around time, concurrent command mode traffic in a video mode transmission, or 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 1-1 shows the predefined and supported display settings, with the following assumptions:

- Clock Lane frequency is 500 MHz that results in a bandwidth of 1 Gbps for each data lane.
- The display should be capable of buffering the pixel data at the speed at which it is delivered in the DSI link.
- No significant control traffic is present on the link when the pixel data is being transmitted.

The last column of Table 1-1 presents the minimum number of lanes required for each configuration.

Table 1-1 should be used only as a reference to evaluate the ability of the link to support the video format for DSI devices.

Table 1-1 Supported Display Settings

Display Type							Video Mode		Command Mode					D-PHY at 1Gbps
Video Interface							DPI Interface		DBI Interface			APB Slave		
	Horizontal		Vertical											
Resolution (pixels)	Active	Total	Active	Total	Refresh Rate (Hz)	Color Depth (bpp)	DSI Bandwidth (Mbps)	DPI Clock (MHz)	Raw DSI Bandwidth (Mbps)[1]	Raw DBI Clock (MHz)[1]	Minimum D-PHY Clock Lane (MHz)[1]	Raw DSI Bandwidth (Mbps)[1]	Raw APB Slave Clock (MHz)[1]	Number of Lanes
320x240	320	380	240	246	60	24	135	5.6	111	6.9	83	111	7	1
640x480	640	692	480	486	60	24	484	20.2	442	27.6	332	442	28	1
800x480	800	830	480	486	55	24	532	22.2	507	31.7	380	507	32	1
800x600	800	1024	600	628	30	24	463	19.3	346	21.6	259	346	22	1
800x600	800	1024	600	628	60	24	926	38.6	N/S[2]	N/S[2]	N/S[2]	691	43	1
960x480	960	1264	480	525	30	24	478	19.9	N/S[2]	N/S[2]	N/S[2]	332	21	1
960x480	960	1264	480	525	60	24	956	39.8	N/S[2]	N/S[2]	N/S[2]	664	41	1
1024x768	1024	1328	768	804	30	24	769	32.0	N/S[2]	N/S[2]	N/S[2]	566	35	1
1024x768	1024	1328	768	804	60	24	1538	64.1	N/S[2]	N/S[2]	N/S[2]	1132	71	2
1280x720	1280	1536	720	732	30	24	810	33.7	N/S[2]	N/S[2]	N/S[2]	664	41	1
1280x720	1280	1536	720	732	60	24	1619	67.5	N/S[2]	N/S[2]	N/S[2]	1327	83	2
1920x1080	1920	2080	1080	1115	30	24	1670	69.6	N/S[2]	N/S[2]	N/S[2]	1493	93	2
1920x1080	1920	2080	1080	1115	60	24	3340	139.2	N/S[2]	N/S[2]	N/S[2]	2986	187	4
1920x1200	1920	2080	1200	1235	30	18	1387	57.8	N/S[2]	N/S[2]	N/S[2]	1244	78	2
1920x1200	1920	2080	1200	1235	30	24	1850	77.1	N/S[2]	N/S[2]	N/S[2]	1659	104	2
1920x1200	1920	2080	1200	1235	60	24	3699	154.1	N/S[2]	N/S[2]	N/S[2]	3318	207	4

1. The values are for the video data only without considering the overhead of commands and FIFO interfaces.
2. N/S: Not Supported due to the speed limitations of DBI interface.

1.6 Clock Speed Requirements

Table 1-2 shows the clock frequencies of the DWC_mipi_dsi_host.

Table 1-2 DWC_mipi_dsi_host Clock Frequencies

Clock Domain	Minimum Frequency	Maximum Frequency
rxclkesc		20 MHz Limited by MIPI D-PHY 1.0 Specification. If required, this value can be exceeded.
txclkesc		20 MHz Limited by MIPI D-PHY 1.0 Specification. If required, this value can be exceeded.
lanebyteclk		125 MHz Limited by MIPI D-PHY 1.0 Specification. If required, this value can be exceeded.
pclk	2 MHz	None (MIPI D-PHY 1.0 Specification does not define the maximum limit) The DWC_mipi_dsi_host 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.
dpipclk		None (MIPI D-PHY 1.0 Specification does not define the maximum limit) The DWC_mipi_dsi_host can support several technologies up to 250 MHz. The maximum speed that can be used depends on the standard cells library that is used for synthesis.
dbiclk		41 MHz 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 interface. It corresponds to the following signals: ■ CSX for type A fixed E interface ■ E for type A clocked E interface ■ WRX/RDX for type B interface

1.7 Area and Power

Table 1-3 shows the power consumption values and 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. The D-PHY

or 2-Port RAM areas are not considered in these numbers. The area is represented by two-input NAND gates.

Table 1-3 DWC_mipi_dsi_host Power and Gate Count

Configuration	Interface	Area (K Gates)	Power in mW
Two data lanes	DBI only	22	0.532
Two data lanes with support for generic packets	DBI only	29.5	0.705
Two data lanes	DPI only	29.9	0.7
Two data lanes with support for generic packets	DPI only	41.9	0.934
Two data lanes	DPI and DBI	43.5	1.03
Two data lanes with support for generic packets	DPI and DBI	52.5	1.21
Two data lanes with support for generic packets	eDPI	46.4	1.11
Four data lanes	DBI only	22.2	0.534
Four data lanes with support for generic packets	DBI only	29.8	0.705
Four data lanes	DPI only	32.3	0.704
Four data lanes with support for generic packets	DPI only	43.8	0.938
Four data lanes	DPI and DBI	45.3	1.03
Four data lanes with support for generic packets	DPI and DBI	54.1	1.21
Four data lanes with support for generic packets	eDPI	48.1	1.12

2 Configuration Parameters

This chapter provides a detailed description of the coreConsultant parameters for the DWC_mipi_dsi_host configuration options in the section "DWC_mipi_dsi_host Configuration Parameters" on page 26.

2.1 DWC_mipi_dsi_host Configuration Parameters

Table 2-1 describes the configuration parameters.

Table 2-1 DWC_mipi_dsi_host Configuration Parameters Descriptions

Configuration Option	Definition
Use SNPS PHY	**Description:** 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: ■ 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> **Dependency:** None **Parameter Name:** DSI_HOST_SNPS_PHY **Values:** ■ 1 (Enabled) ■ 0 (Disabled) **Default Value:** 0 (Disabled)
Select the number of synchronization stages	**Description:** Specifies the number of synchronization stages for clock domain crossing. All the stages capture the data on the rising edge of the clock. **Dependency:** None **Parameter Name:** DSI_HOST_DFLT_F_SYNC_TYPE **Values:** 2, 3, or 4 **Default Value:** 2
Select the number of interrupt pins	**Description**: Specifies the maximum number of interrupt pins. **Dependency:** None **Parameter Name:** DSI_N_INTERRUPT_PINS **Values:** ■ 0: None ■ 1: One interrupt for INT_ST0 and INT_ST1 registers ■ 2: Two interrupts for INT_ST0 and INT_ST1 registers **Default Value:** 2
Select the number of lanes supported by D-PHY	**Description**: Specifies the maximum number of data lanes in the design. **Dependency:** None **Parameter Name:** DSI_HOST_NUMBER_OF_LANES **Values:** 1, 2, 3, or 4 **Default Value:** 2

Table 2-1 DWC_mipi_dsi_host Configuration Parameters Descriptions (Continued)

Configuration Option	Definition
Use SNPS D-PHY 4L Daughter Card	**Description**: Arranges the design in a hierarchical structure that is specific for FPGA prototyping. **Dependency:** This option is available only if the D-PHY is external and has four lanes (DSI_HOST_SNPS_PHY = 0 and DSI_HOST_NUMBER_OF_LANES = 4) **Parameter Name:** DSI_HOST_FPGA **Values:** ■ 1 (Enabled) ■ 0 (Disabled) **Default Value:** 0 (Disabled)
Select the system interface	**Description**: Configures the system data interface. Different memory models are required to support these interfaces. **Dependency:** None **Parameter Name:** DSI_DATAINTERFACE **Values:** ■ DBI: MIPI Display Bus Interface ■ DPI: MIPI Display Pixel Interface ■ DBI&DPI: Both DBI and DPI interfaces ■ EDPI: Enhanced Display Bus Interface **Default Value:** DBI&DPI
Select the DBI interface type	**Description**: Configures the type of DBI system interface. **Dependency:** This option is available when you select DBI or DBI&DPI as the system interface (DSI_DATAINTERFACE = DBI or DBI&DPI). **Parameter Name:** DSI_DBIINTERFACE_TYPE **Values:** ■ 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 **Default Value:** TYPE_A_CLOCKED_E
Enable the support for Generic Packets	**Description**: Enables the support for the Generic packets that are conveyed through the AMBA APB interface. Memory models are required to support this feature. **Dependency:** This option is available when you select a system interface other than EDPI (DSI_DATAINTERFACE != EDPI). **Parameter Name:** DSI_GENERIC **Values:** ■ 1 (Enabled) ■ 0 (Disabled) **Default Value:** 1 (Enabled)

Table 2-1 DWC_mipi_dsi_host Configuration Parameters Descriptions (Continued)

Configuration Option	Definition
Select the type of 2-port RAM memory	**Description**: Configures the memory type to which the FIFO controllers connect. **Dependency:** This option is available only if FGPA prototyping is disabled (DSI_HOST_FPGA = 0) **Parameter Name:** DSI_HOST_SNPS_SYNC_RD_FIFOS **Values:** ■ Asynchronous ■ Synchronous **Default Value:** Synchronous
DPI Payload FIFO Depth (uses 2-port RAM)	**Description**: Specifies the DPI Payload 2-port RAM address depth in 32-bit slots. It requires a 2-port RAM model. You should define the DPI FIFO memory depth such that it is sufficient to store at least one entire line of video with the maximum resolution that you want to support. For example, consider supporting Full HD 1080p, where one video line contains 1920 pixels. If you use 24 bpp, each line contains 1920 x 24 = 46080 bits. The DPI pixel memory stores 32 bits per memory slot. Therefore, you should define a minimum of 46080/32 = 1440 for the DPI FIFO memory depth. 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. **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. **Dependency:** This option is available when you select DPI, DBI&DPI, or EDPI as the system interface (DSI_DATAINTERFACE = DPI, DBI&DPI, or EDPI). **Parameter Name:** DSI_PIXELMEMADDRDEPTH **Values:** 8 to 4096 **Default Value:** 488

Table 2-1 DWC_mipi_dsi_host Configuration Parameters Descriptions (Continued)

Configuration Option	Definition
DBI Command FIFO Depth (uses Flip-Flops)	**Description**: Specifies the DBI Command FIFO address depth in 32-bit slots. The FIFO is implemented using flip-flops. You should define the DBI Command FIFO depth 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 DBI interface and resume normal procedure, the DBI Command FIFO depth should be configured with the minimum size of that burst. Consider that the initialization sequence of a given display consists of 10 sequential commands. In this case, you should define a minimum of 10 for the DBI Command FIFO depth. **Dependency:** This option is available when you select DBI or DBI&DPI as the system interface (DSI_DATAINTERFACE = DBI or DBI&DPI). **Parameter Name:** DSI_DBICMDADDRDEPTH **Values:** 8 to 1024 **Default Value:** 10
DBI Payload FIFO Depth (uses 2-port RAM)	**Description**: Specifies the DBI Payload 2-port RAM address depth in 32-bit slots. It requires a 2-Port RAM model. You should define the DBI Write Payload FIFO depth such that it is sufficient to store the entire payload content of the number of packets defined in the DBI Command FIFO depth. 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 x 40 = 400 bytes. Each FIFO memory slot stores 4 bytes. Therefore, you should define a minimum of 400/4 = 100 for the DBI Write Payload FIFO depth. **Dependency:** This option is available when you select DBI or DBI&DPI as the system interface (DSI_DATAINTERFACE = DBI or DBI&DPI). **Parameter Name:** DSI_DBIPLDADDRDEPTH **Values:** 49 to 1024 **Default Value:** 50

Table 2-1 DWC_mipi_dsi_host Configuration Parameters Descriptions (Continued)

Configuration Option	Definition
DBI Read FIFO Depth (uses 2-port RAM)	**Description**: Specifies the DBI Read 2-Port RAM address depth in 32-bit slots. It requires a 2-Port RAM model. You should define the DBI Read Payload FIFO depth 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. Each FIFO memory slot stores 4 bytes. Therefore, you should define a minimum of 40/4 = 10 for the DBI Read Payload FIFO depth. **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. **Dependency**: This option is available when you select DBI or DBI&DPI as the system interface (DSI_DATAINTERFACE = DBI or DBI&DPI). **Parameter Name**: DSI_DBIREADPLDADDRDEPTH **Values**: 8 to 1024 **Default Value**: 20
Generic Command FIFO Depth (uses Flip-Flops)	**Description**: Specifies the Generic Command FIFO address depth in 32-bit slots. The FIFO is implemented using flip-flops. You should define the Generic Command FIFO depth 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 FIFO depth should be configured with the minimum size of that burst. Consider that the initialization sequence of a given display consists of 10 sequential commands. In this case, you should define a minimum of 10 for the Generic Command FIFO depth. **Dependency:** This option is available when you enable the support for Generic packets (DSI_GENERIC = 1). **Parameter Name:** DSI_GENERICCMDADDRDEPTH **Values:** 8 to 1024 **Default Value:** 10

Table 2-1 DWC_mipi_dsi_host Configuration Parameters Descriptions (Continued)

Configuration Option	Definition
Generic Payload FIFO Depth (uses 2-port RAM)	**Description**: Specifies the Generic Payload 2-Port RAM address depth in 32-bit slots. It requires a 2-Port RAM model. You should define the Generic Write Payload FIFO depth such that it is sufficient to store the entire payload content of the number of packets defined in the Generic Command FIFO depth. 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 x 40 = 400 bytes. Each FIFO memory slot stores 4 bytes. Therefore, you should define a minimum of 400/4 = 100 for the Generic Write Payload FIFO depth. **Dependency:** This option is available when you enable the support for Generic packets (DSI_GENERIC = 1). **Parameter Name:** DSI_GENERICPLDADDRDEPTH **Values:** 8 to 1024 **Default Value:** 200
Generic Read FIFO Depth (uses 2-port RAM)	**Description**: Specifies the Generic Read 2-Port RAM address depth in 32-bit slots. It requires a 2-Port RAM model. You should define the Generic Read Payload FIFO depth 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. Each FIFO memory slot stores 4 bytes. Therefore, you should define a minimum of 40/4 = 10 for the Generic Read Payload FIFO depth. **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. **Dependency:** This option is available when you enable the support for Generic packets (DSI_GENERIC = 1). **Parameter Name:** DSI_GENREADPLDADDRDEPTH **Values:** 8 to 1024 **Default Value:** 16

Note Synopsys recommends that after generating a configuration, you use it in a system simulation to ensure that, in the worst case environment, the controller FIFO depth definitions are sufficient for your scenario demands. You should define the DWC_mipi_dsi_host FIFOs depending on the packet that is generated by each interface.

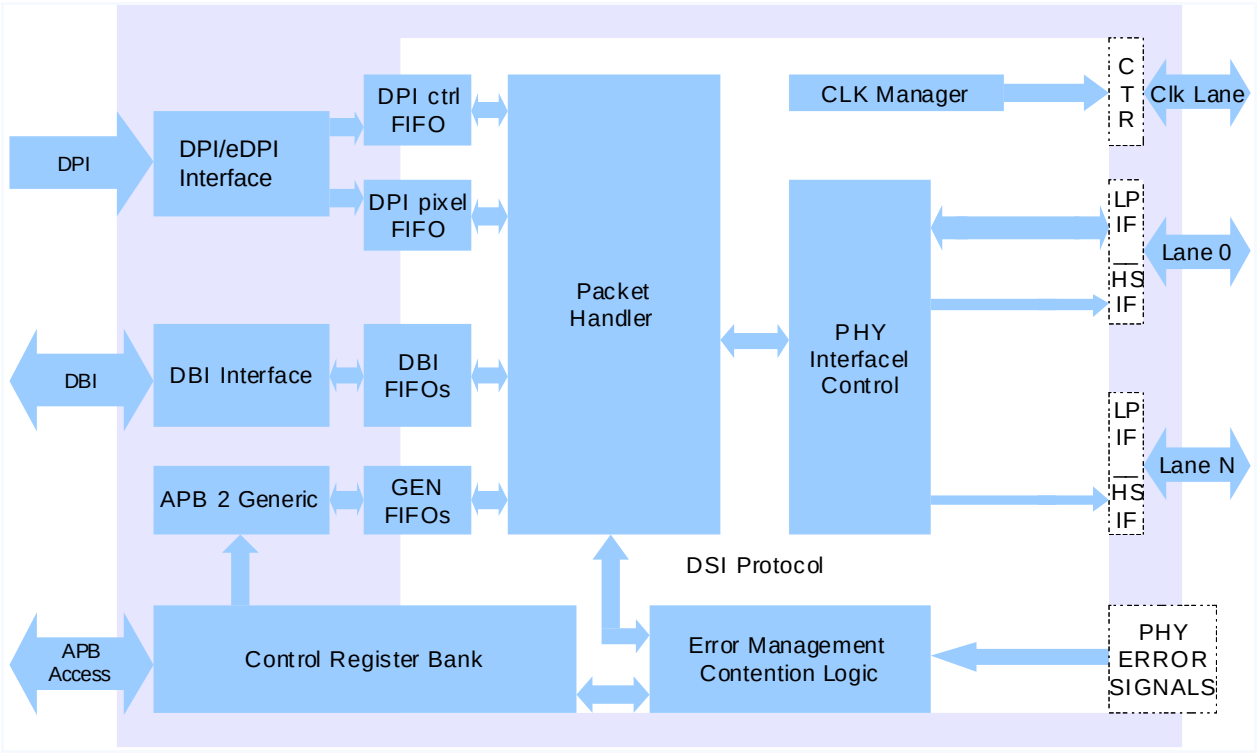
3 Architecture

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

3.1 System Level Architecture

Figure 3-1 shows the architecture of DWC_mipi_dsi_host.

Figure 3-1 Architecture of DWC_mipi_dsi_host



The different parts have the following functions:

- The DPI/eDPI interface captures the data and control signals and conveys them to a FIFO for video control signals and another one for the pixel data. This data is then used to build one of the following:
  - Video packets, when in Video mode (see "DPI Interface" on page 51)
  - The memory_write_start and memory_write_continue DCS commands, when in Command mode (see "eDPI Interface" on page 62)
- The DBI interface encapsulates the DCS commands in DSI packets that are then conveyed to the command and payload FIFOs. For commands that require a response from the device, the block uses an incoming data FIFO to acquire data from a peripheral. For more information about the DBI interface, see "DBI Interface" on page 44.
- The Register Bank is accessible through a standard AMBA-APB slave interface, providing access to the DWC_mipi_dsi_host registers for configuration and control. There is also a fully programmable interrupt generator to inform the system about certain events.
- The PHY Interface Control is responsible for managing the D-PHY PPI interface. It acknowledges the current operation and enables low-power transmission/reception or a high-speed transmission. It also performs data splitting between available D-PHY lanes for high-speed transmission.

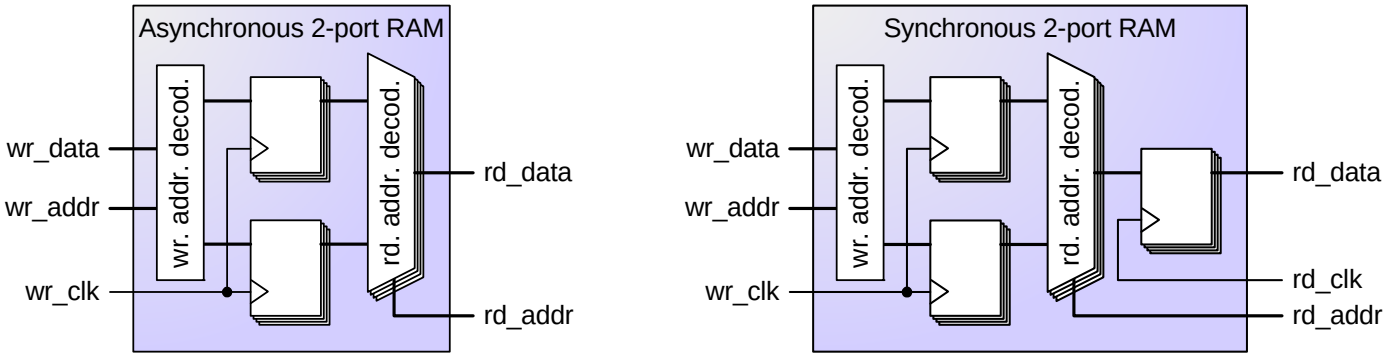
- The Packet Handler schedules the activities inside the link. It performs several functions based on the interfaces that are currently operational (DPI or DBI) and the video transmission mode that is used (burst mode or non-burst mode with sync pulses or sync events). It builds long or short packet generating correspondent ECC and CRC codes. This block also performs the following functions:
  - Packet reception
  - Validation of packet header by checking the ECC
  - Header correction and notification for single-bit errors
  - Termination of reception
  - Multiple header error notification

  Depending on the virtual channel of the incoming packet, the handler routes the output data to the respective port (Generic or DBI).
- The APB-to-Generic block bridges the APB operations into FIFOs holding the Generic commands. The block interfaces with the following FIFOS:
  - Command FIFO
  - Write payload FIFO
  - Read payload FIFO
- The Error Management notifies and monitors the error conditions on the DSI link. It controls the timers used to determine if a timeout condition occurred, performing an internal soft reset and triggering an interruption notification.

3.2 Supported 2-Port RAM Types

The 2-port RAMs are used by the DWC_mipi_dsi_host to implement FIFOs for data storage and synchronization. The 2-port RAMs are divided into two categories: synchronous and asynchronous. These names are used to distinguish the way data is accessed for read operations, as shown in Figure 3-2.

Figure 3-2 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 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. You can select the type of RAM during the core configuration in coreConsultant using the option Select the type of 2-port RAM memory.

3.3 Timeout Timers

3.3.1 Peripheral Response Timeout

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

Table 3-1 lists the events belonging to various categories having an associated timeout for peripheral response.

Table 3-1 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 HS and LP)	■ (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 HS and LP)	■ (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

TheDWC_mipi_dsi_host ensures that, on sending an event that triggers a timeout, the D-PHY switches to the Stop state and a counter starts running until it reaches the value of that timeout. The link remains in the LP-11 state and unused until the timeout ends, even if there are other events ready to be transmitted.

Figure 3-3, Figure 3-4, and Figure 3-5 illustrate the flow of counting in the PRESP_TO counter for the three categories listed in Table 3-1.

Figure 3-3 Timing of PRESP_TO after a Bus Turn-Around

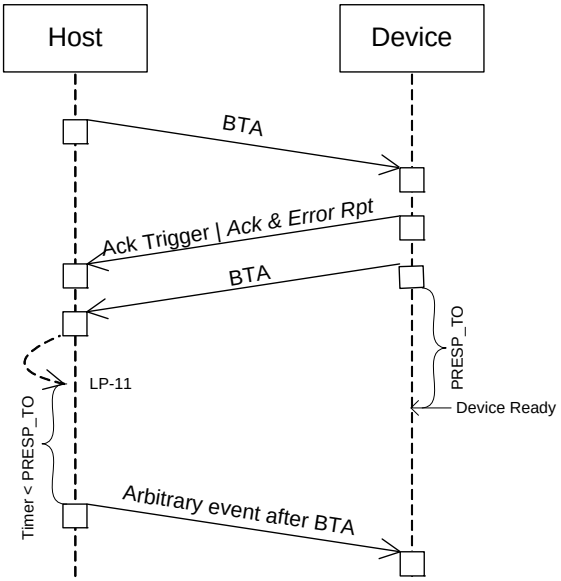


Figure 3-4 Timing of PRESP_TO after a Read Request (HS or LP)

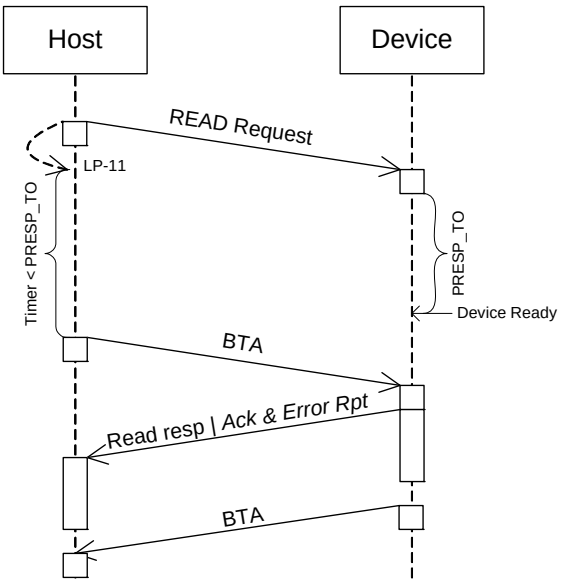
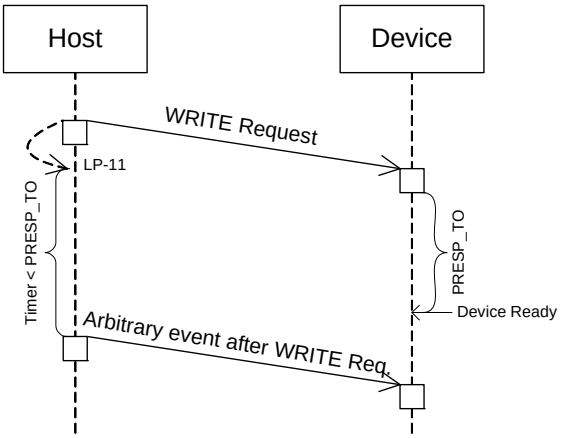


Figure 3-5 Timing of PRESP_TO after a Write Request (HS or LP)



The BTA_TO_CNT, HS_RD_TO_CNT, LP_RD_TO_CNT, HS_WR_TO_CNT, and LP_WR_TO_CNT registers are used for the configuration of the PRESP_TO counter.

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

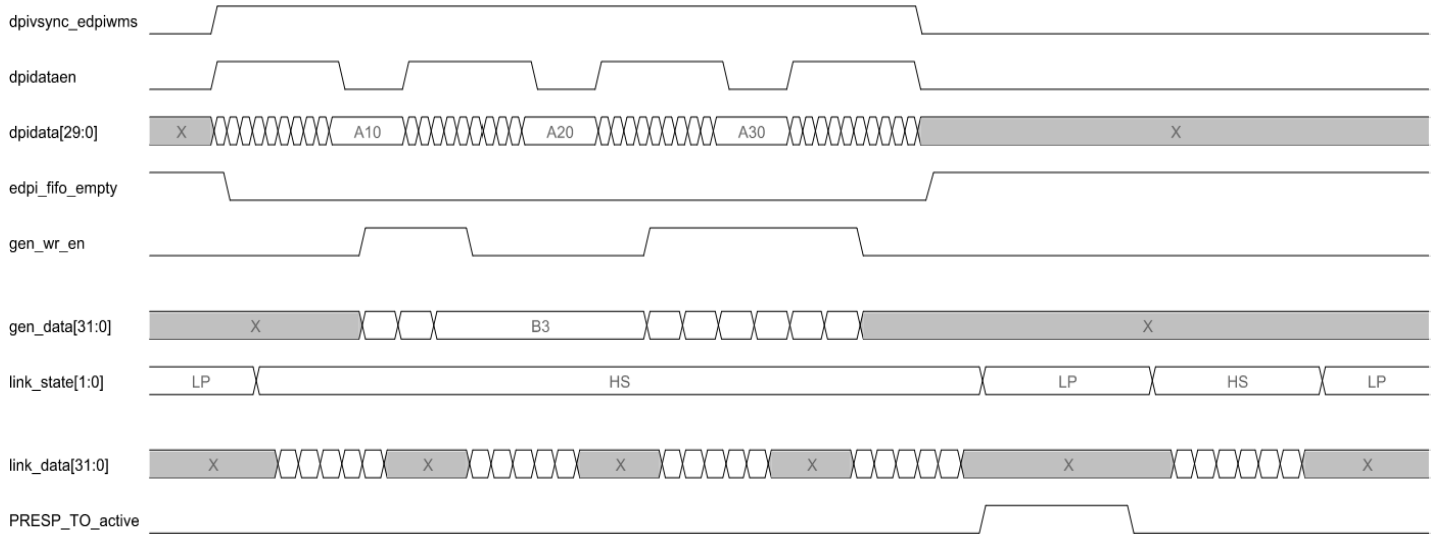
The read and the write requests in high-speed mode are distinct from the read and the write requests in low-power mode. For example, if HS_RD_TO_CNT is set to zero and LP_RD_TO_CNT is set to a non-zero value, a generic read with no parameters does not activate the PRESP_TO counter in high-speed, but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

The HS_WR_TO_CNT register includes a special bit, presp_to_mode, to change the normal behavior of PRESP_TO for the eDPI interface. When set to 1, this bit allows the PRESP_TO from HS_WR_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 interface are transmitted and its FIFO is empty again.

In this scenario, non-eDPI requests are not sent to the D-PHY, even if there is traffic from the Generic or the DBI interfaces ready to be sent, returning them to the Stop state. When it happens, the PRESP_TO counter is activated and only when it is completed, the DWC_mipi_dsi_host sends any other traffic that is ready. This is illustrated in Figure 3-6.

Figure 3-6 Effect of presp_to_mode at 1



3.3.2 Errors Raised by the Timeouts

The DWC_mipi_dsi_host implements a set of timers and conditions to notify the errors. The DWC_mipi_dsi_host has a set of registers to control the timers that are used to determine if a timeout has occurred. It also contains a set of interruption status registers that are cleared upon read operation. Optionally, these registers also trigger an interrupt signal that can be used by the system to be activated when an error occurs within the DSI connection.

The to_hs_tx field in the INT_ST1 register is set when a high-speed transmission timeout occurs 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.

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

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 as input. This mechanism increases the range to define these limits.

3.4 Error Control

The INT_ST0 and INT_ST1 registers are associated with error condition reporting. These registers can trigger interrupt pins to inform the system about the occurrence of errors. The coreConsultant allows the following interrupt pin configurations:

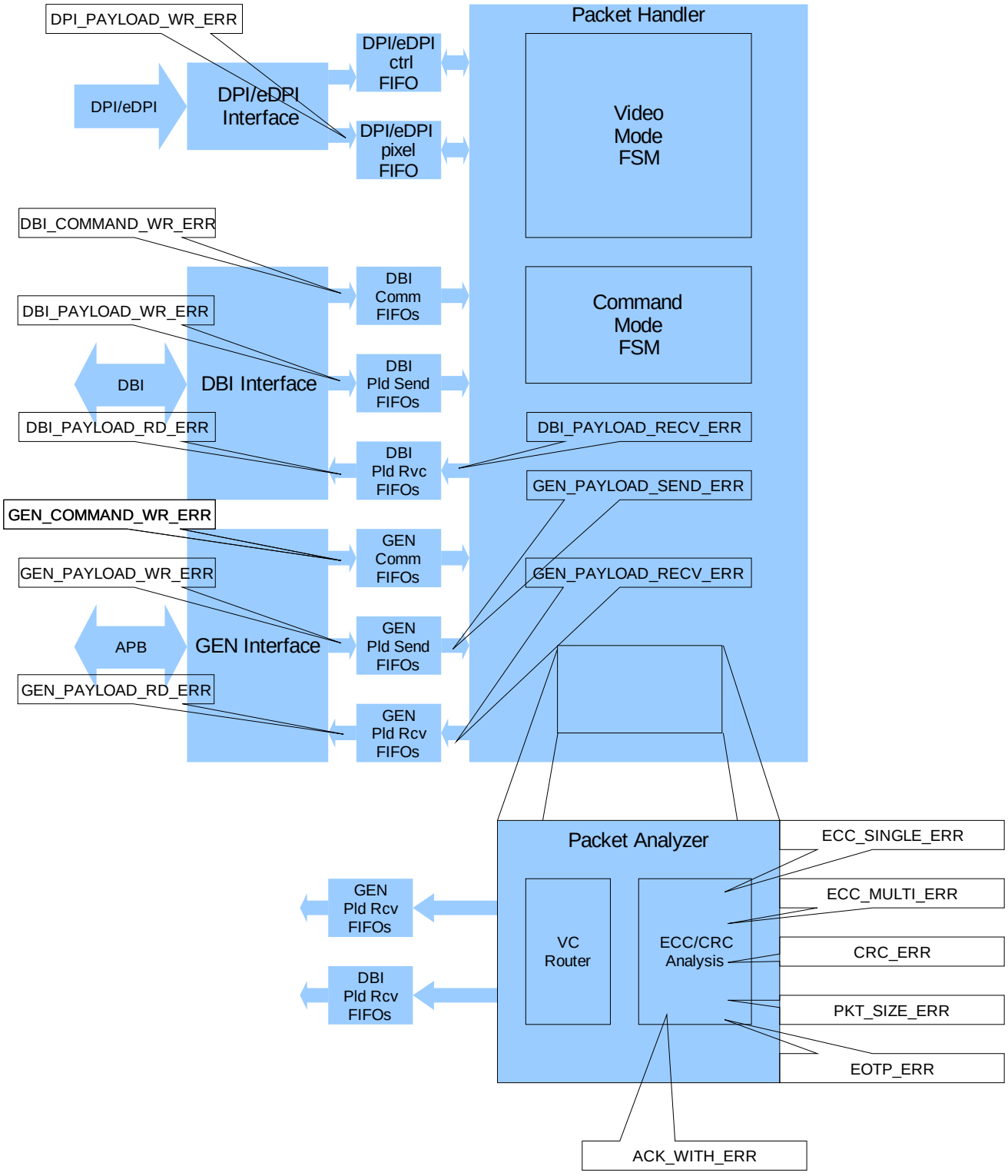
- The DWC_mipi_dsi_host does not have any interrupt pin for notification of error, although the system can check for error by reading the error status registers, INT_ST0 and INT_ST1.

- The DWC_mipi_dsi_host has one interrupt pin interrupt that is set high when an error occurs either in INT_ST0 and INT_ST1 register.
- The DWC_mipi_dsi_host has two interrupts associated with each error status register. The interrupt0 and interrupt1 pins are respectively associated with INT_ST0 and INT_ST1 registers. This mechanism allows faster association of an interrupt with the status register that generates the error.

The triggering of the interrupt pins, when defined, can be masked by programming the mask registers INT_MSK0 and INT_MSK1. When any bit of these registers is set to 1, it inhibits the interrupt for a specific error. The error bit is always set in the respective INT_ST register. The INT_ST0 and INT_ST1 registers are always cleared after a read operation.

Figure 3-7 on page 41 illustrates the location of some of the errors. For information about the causes of the errors and the action to be taken see "Error Handling" on page 191.

Figure 3-7 Error Sources



4 Video Interfaces

The DWC_mipi_dsi_host includes several video and configuration interfaces to facilitate integration with the host systems and for compatibility with different displays. This chapter describes these interfaces and their functionalities in the following sections:

4.1 DBI Interface

The DBI interface follows the MIPI DBI specification. It is used to transmit 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 supports type A and type B variants of this interface. The DBI interface supports the Command mode devices where the transactions are carried out through the commands as defined in the DCS specification.

The DBI interface does not operate concurrently with DPI interface.

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

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

You can select one of the following DBI interfaces using the Select the DBI interface type option:

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

The Type C serial mode is not supported. The selection type maps the related pins on the DWC_mipi_dsi_host pinout. The DBI interface uses RGB additive color mixing method, and the pixel data format is selected in 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 should be associated with the DSI Virtual Channel that is configured in the dbi_vcid field.

See Appendix A, "DBI Color Code Mapping Waveforms" for the diagrams on pixel-to-byte conversion for each mode when the interface works as type A with a fixed E.

4.1.1 DSI Packet Types

The DBI interface receives the commands and decodes the information it needs for the command packetizing format. Depending on the command type, the DWC_mipi_dsi_host outputs the related DSI packet type (DCS short write packet, DCS read packet, or DCS long write). If the DCS commands have variable sizes, configure the sizes in the DBI_CMDSIZE register. You can configure the DBI_CMDSIZE register only when the DWC_mipi_dsi_host is idle and not transmitting. Table 4-1 shows the DSI packet types for DCS commands.

Table 4-1 DSI Packet Types for DCS Commands

DCS Command	Command ID	Number Of Parameters	DSI Packet Type
enter_idle_mode	0x39h	0	6'h05
enter_invert_mode	0x21h	0	6'h05
enter_normal_mode	0x13h	0	6'h05
enter_partial_mode	0x12h	0	6'h05

Table 4-1 DSI Packet Types for DCS Commands (Continued)

DCS Command	Command ID	Number Of Parameters	DSI Packet Type
enter_sleep_mode	0x10h	0	6'h05
exit_idle_mode	0x38h	0	6'h05
exit_invert_mode	0x20h	0	6'h05
exit_sleep_mode	0x11h	0	6'h05
nop	0x00h	0	6'h05
set_display_off	0x28h	0	6'h05
set_display_on	0x29h	0	6'h05
set_tear_off	0x34h	0	6'h05
soft_reset	0x01h	0	6'h05
set_address_mode	0x36h	1	6'h15
set_gamma_curve	0x26h	1	6'h15
set_pixel_format	0x3Ah	1	6'h15
set_tear_on	0x35h	1	6'h15
set_scroll_start	0x37h	2	6'h39
set_tear_scanline	0x44h	2	6'h39
set_column_address	0x2Ah	4	6'h39
set_page_address	0x2Bh	4	6'h39
set_partial_rows	0x30h	4	6'h39
set_partial_columns	0x31h	4	6'h39
set_scroll_area	0x33h	6	6'h39
write_LUT	0x2Dh	Variable	6'h39
write_memory_continue	0x3Ch	Variable	6'h39
write_memory_start	0x2Ch	Variable	6'h39
get_address_mode	0x0Bh	1(read)	6'h06
get_blue_channel	0x08h	1(read)	6'h06
get_diagnostic_result	0x0Fh	1(read)	6'h06
get_display_mode	0x0Dh	1(read)	6'h06
get_green_channel	0x07h	1(read)	6'h06

Table 4-1 DSI Packet Types for DCS Commands (Continued)

DCS Command	Command ID	Number Of Parameters	DSI Packet Type
get_pixel_format	0x0Ch	1(read)	6'h06
get_power_mode	0x0Ah	1(read)	6'h06
get_red_channel	0x06h	1(read)	6'h06
get_scan_line	0x45h	1(read)	6'h06
get_signal_mode	0x0Eh	1(read)	6'h06
read_DDB_continue	0xA8h	1(read)	6'h06
read_DDB_start	0xA1h	1(read)	6'h06
read_memory_continue	0x3Eh	1(read)	6'h06
read_memory_start	0x2Eh	1(read)	6'h06
set_3d_control	0x06h	2	6'h06
get_3d_control	0x39h	0	6'h06
set_vsync_timing	0x15h	1	6'h06

Commands with zero or one parameter are encapsulated in short packets and commands with more than one parameter are encapsulated in long packets.

4.1.2 Configuring the DBI Pixel Data

There are several register fields to configure the DBI input pixel data and the DBI pixel data retrieval. Prior knowledge of the supported display feature is required to avoid incorrect core configuration.

The in_dbi_conf field in the DBI_CFG register configures the color depth/pixel format used by the input interface. The out_dbi_conf field configures the color depth/pixel format used by the output interface. This configuration is directly related with the formats supported by the peripheral. If the peripheral returns pixel data with a specific color depth/pixel format, this should be consistent with the out_dbi_conf register configuration. If the format is not followed, the byte-to-pixel converter uses an incorrect pixel mapping and corrupts output data.

The out_dbi_conf field should not be programmed for a higher number of interface pins than in_dbi_conf. For example, when in_dbi_conf defines a 9-bit interface and out_dbi_conf defines a 16-bit interface, an incorrect pin configuration occurs because pins 16 to 10 are not available.

The out_dbi_conf field configures the size of the DCS write_LUT command. This command programs the color space conversion table of the peripheral. This register is directly related with the capabilities of the peripheral. If the peripheral supports a color depth of 16/18/24 bits per pixel, it is expected that this register is configured for 48/128/192 bytes, respectively. These coefficients should be loaded to the peripheral before normal operation of pixel data transmission is started. The DCS specification only refers to the conversion of formats equal or greater than 12 bpp. This means that any format outside this scope may not be convertible to a higher color depth mode by the peripheral. Taking this into consideration, when in_dbi_conf selects a color depth less than 12 bpp, the out_dbi_conf field should be configured such that the

color depth does not have a higher value. For example, if in_dbi_conf is configured for 8 bits/8 bpp interface and out_dbi_conf is configured for 8 bits/12 bpp, this results in an incorrect configuration because the DCS specification does not allow conversions from 8 bpp to 12 bpp. This configuration is possible inside the core but may lead to incorrect results. However, the system should configure these registers according to the desired operation.

The partitioning_en field of the DBI_PARTITIONING_EN register configures the option to be the processor itself to perform the long packet write_memory_start division, issuing write_memory_continue commands on its own, as appropriate. Otherwise, if this bit is not enabled, the DWC_mipi_dsi_host splits a write_memory_start command into one write_memory_start and several write_memory_continue commands, as necessary.

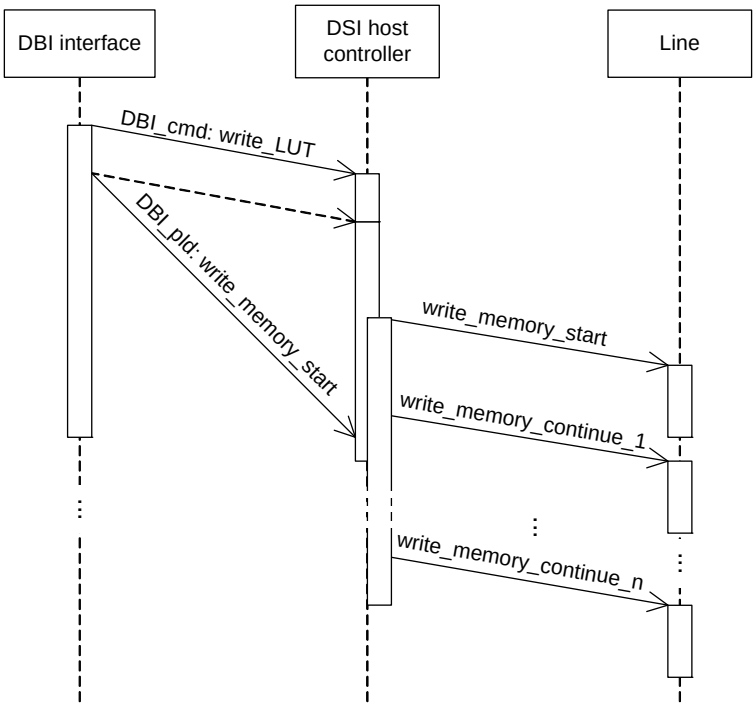
For Read Back commands, the DBI specification states that the reading process does not use any acknowledge mechanism to know that the requested data is ready. The system can get this information by polling the Command Status Register (CMD_PKT_STATUS). This register, besides other information, contains the status on the DBI read back payload FIFO and also contains a special busy flag (dbi_rd_cmd_busy) that can be used to understand if the read command is complete. By using the DBI read back FIFO status bits, that report if the FIFO is full or empty, the processor can understand if it can read another byte/sample from the interface (probing the empty flag) or when it should stop reading bytes/samples from the interface.

Additionally, the Read Command Busy flag is asserted when a read command is issued by the interface and cleared when the reading process ends and the read data is fully stored inside the read back FIFO. After issuing a read command, the system can periodically probe this status bit to acknowledge if the read back operation is complete. When this information is available, the system can assume that the entire data is stored inside the read back FIFO and the interface can deliver all the desired data.

4.1.3 Partitioning the Long write_memory_start Commands

The DWC_mipi_dsi_host can divide a long DBI write_memory_start command into several packets of shorter length, consisting of one write_memory_start packet followed by several write_memory_continue packets. This partitioning is illustrated in Figure 4-1.

Figure 4-1 Transmission of a Long write_memory_start Packet Partitioned by DSI Host Controller



This feature allows the DBI interface to issue multiple command packets to be transmitted to the DSI link. The initial write_memory_start packet and each write_memory_continue packet generated by the DWC_mipi_dsi_host are bonded to a DSI packet. The packet is limited in size by the value programmed in the allowed_cmd_size field of the DBI_CMDSIZE register.

The wr_cmd_size field configures the full size of the DCS memory write command. As an example, if allowed_cmd_size = 200 and wr_cmd_size = 900, the DWC_mipi_dsi_host sends five memory write DCS commands. The first four commands have a length of 200 bytes and the last command conveys the remaining 100 bytes. The values in these fields are measured in bytes.

Table 4-2 shows the conversion of the number of pixels to bytes that should 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 has to 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 even number of pixels, because in 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 4-2 Number of Pixels to Bytes Conversion for the DBI Pixels Formats

DBI Interface	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) +1[1]	(3*number of pixels/2) +1[1]
16-bit 16 bpp	1000	(2*number of pixels) + 1	(2*number of pixels) +1

Table 4-2 Number of Pixels to Bytes Conversion for the DBI Pixels Formats (Continued)

DBI Interface	DBI_CFG Register	DBI_CMDSIZE Register	
Pixel Data Format	**in_dbi_conf Field**	**wr_cmd_size Field**	**allowed_cmd_size Field**
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) +1[1]	(3*number of pixels/2) +1[1]
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.

4.1.4 Handling the Low Transmission Speed

If the transmission speed is slow (for example, LP transmission), the commands at the DBI interface might be generated faster than they are dispatched on the DSI link, causing the command FIFO to overflow. This happens because the DBI interface does not verify if the DBI command FIFO is full. The data is lost and unrecovered, and the system needs to issue a soft reset to recover.

To avoid this overflow situation, the system should 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 simultaneously avoids FIFO starvation, making it possible to use FIFO sizes smaller than the amount of data of the longest packet to be written.

4.1.5 Dimensioning the Size of FIFOs for Automatic Partitioning

For the DWC_mipi_dsi_host configuration using automatic partitioning, the size of the FIFOs should be calculated considering that there may be some wastage due to the organization of data.

For example, if the allowed_cmd_size packet is set 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 is not packed together in memory. Therefore, there is a wastage of 2 bytes of storage space per packet in this case. A schematic diagram of the memory contents for this example is shown in Figure 4-2.

Figure 4-2 Example of Data Storage in DBI Payload FIFO

Automatic partition imposes minimum sizes for FIFOs when you send a total payload size of pld_size bytes when the maximum allowed size per packet is set to max_allowed bytes. This minimum size is calculated using the following expressions:

$$min\_size\_for\_DBI\_cmd\_FIFO = \left\lceil \frac{pld\_size}{max\_allowed} \right\rceil$$

$$min\_size\_for\_DBI\_pld\_FIFO = \left\lceil \frac{max\_allowed}{4} \right\rceil . min\_size\_for\_DBI\_cmd\_FIFO$$

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

For the example in Figure 4-2:

- pld_size = 24 bytes
- max_allowed = 6 bytes

Therefore,

$$min\_size\_for\_DBI\_cmd\_FIFO = \left\lceil \frac{24}{6} \right\rceil = \lceil 4 \rceil = 4$$

$$min\_size\_for\_DBI\_pld\_FIFO = \left\lceil \frac{6}{4} \right\rceil . 4 = \lceil 1.5 \rceil . 4 = 2 . 4 = 8$$

Figure 4-2 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.

4.2 DPI Interface

The DPI interface follows the MIPI DPI-2 specification with pixel data bus width up to 24 bits. It is used to transmit the information in Video mode in which the transfers from the host processor to the peripheral take the form of a real-time pixel stream. This interface allows sending ShutDown (SD) and ColorMode (CM) commands, which are triggered directly by the pins at the interface. To transfer additional commands (for example, to initialize the display), use another interface such as APB Slave Generic Interface to complement the DPI interface.

Note The DPI interface does not operate concurrently with the DBI interface.

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

The DPI interface has a pin that is not included in the DPI-2 specification. The edpihalt pin is added to the DPI interface to aid the DPI video driver to control the input rate of data when the DWC_mipi_dsi_host is executing an instruction and cannot accept additional pixel data. For more information about the behavior of edpihalt, see "Transmission of Commands" on page 77.

The DPI interface captures the data and control signals and conveys them to the FIFO interfaces that transmit them to the DSI link. Two different streams of data are presented at the interface; video control signals and pixel data. Depending on the interface color coding, the pixel data is disposed differently throughout the dpipixdata bus. Interface pixel color coding is shown in Figure 4-3.

The DPI interface can be configured to increase flexibility and promote correct usage of this interface for several systems. These configuration options are as follows:

- Polarity control: All the control signals are programmable to change the polarity depending on system requirements.
- After the core reset, DPI waits for the first VSYNC active transition to start signal sampling, including pixel data, thus avoiding starting the transmission of the image data in the middle of a frame.
- If interface pixel color coding is 18 bits and the 18-bit loosely packed stream is disabled, the number of pixels programmed in the vid_pkt_size field must be a multiple of four. This means that in this mode, the two LSBs in the configuration are always inferred as zero. The specification states that in this mode, the pixel line size should be a multiple of four.
- To avoid FIFO underflows and overflows, the configured number of pixels is assumed to be received at all times. This happens even if the dpidataen pin is active for more or less time than necessary.
- To keep the memory organized with respect to the packet scheduling, the number of pixels per packet parameter is used to separate the memory space of different video packets.

For dpishutdn and dpicolorm sampling and transmission, the DPI video signalling must be active. This means that if the DPI commands are not actively generating the video signals like VSYNC and HSYNC, these signals are not transmitted through the DSI link. Because of such constraints and for commands to be correctly transmitted, the first VSYNC active pulse should occur for the command sampling and transmission. When shutting down the display, it is necessary for the DPI video to keep active for one frame after the command being issued. This ensures that the commands are correctly transmitted before actually disabling the video generation at the DPI interface.

The MIPI DPI-2 specification does not define a standard way to get the data of certain 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 4-3 on page 54.

For all of the data types except the 36-bit format and 12-bit YCbCr 4:2:0, one entire pixel is received per clock cycle. The exceptions are as follows:

- For 36-bit format, receiving one entire pixel per clock cycle on a continuous basis is not possible because this data is stored in a RAM which is only 32-bit wide, using the same clock. Therefore, half a pixel is received in each clock cycle, 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 restrictions to the number of bytes per packet 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 4-3 Location of Color Components in an Extension of DPI-2 Pixel Interface

	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-Bit
Order	1° Cycle	2° Cycle	1° Cycle	2° Cycle	1° Cycle	2° Cycle		1° Cycle	2° 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]
D22	Y1[8]	Y2[8]	Y1[10]	Y2[10]	Y1[6]	Y2[6]	R[2]			Y1[6]	Y1[6]						R[6]
D21	Y1[7]	Y2[7]	Y1[9]	Y2[9]	Y1[5]	Y2[5]	R[1]			Y1[5]	Y1[5]			R[4]		R[5]	R[5]
D20	Y1[6]	Y2[6]	Y1[8]	Y2[8]	Y1[4]	Y2[4]	R[0]			Y1[4]	Y1[4]		R[4]	R[3]		R[4]	R[4]
D19	Y1[5]	Y2[5]	Y1[7]	Y2[7]	Y1[3]	Y2[3]	G[9]			Y1[3]	Y1[3]		R[3]	R[2]		R[3]	R[3]
D18	Y1[4]	Y2[4]	Y1[6]	Y2[6]	Y1[2]	Y2[2]	G[8]			Y1[3]	Y1[3]		R[2]	R[1]		R[2]	R[2]
D17	Y1[3]	Y2[3]	Y1[5]	Y2[5]	Y1[1]	Y2[1]	G[7]	R[11]	G[11]	Y1[1]	Y1[1]		R[1]	R[0]	R[5]	R[1]	R[1]
D16	Y1[2]	Y2[2]	Y1[4]	Y2[4]	Y1[0]	Y2[0]	G[6]	R[10]	G[10]	Y1[0]	Y1[0]		R[0]		R[4]	R[0]	R[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]
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]
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]
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]
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]
D10	Cb1[8]	Cr2[8]	Cb1[10]	Cr2[10]	Cb1[6]	Cr2[6]	G[0]	R[4]	B[10]	Y0[3]	Y0[3]	G[5]	G[2]	G[2]	G[4]	G[2]	G[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]
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]
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]
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]
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]
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[3]	B[4]	B[4]	B[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[2]	B[3]	B[3]	B[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[1]	B[2]	B[2]	B[2]
D1			Cb1[1]	Cr2[1]			B[1]	G[1]	B[1]	Cb[1]	Cr[1]	B[1]	B[1]	B[0]	B[1]	B[1]	B[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]

The number of pixels of payload is restricted to be a multiple of a value, according to Table 4-3. 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 DWC_mipi_dsi_host automatically fills the remaining bits.

Table 4-3 Multiplicity of the Payload Size in Pixels for Each Data Type

Value	Data Types
1	■ 16-bit ■ 18-bit loosely packed ■ 24-bit ■ 30-bit ■ 36-bit
2	■ 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	■ 18-bit non-loosely packed

4.2.1 Configuring the DPI Resolution for Transmission

Perform the following steps to configure the DPI packets transmission:

1. Global configuration:

   Configure n_lanes (PHY_IF_CFG) to define the number of lanes with which the controller can perform high-speed transmissions.

2. Configure the DPI Interface to define how the DPI interface interacts with the controller.

   - Configure dpi_vcid (DPI_VCID): This field configures the virtual channel that the packets generated by the DPI interface is indexed to.
   - Configure dpi_color_coding (DPI_COLOR_CODING): This field configures the bits per pixel that the interface transmits and also the variant configuration of each bpp. If you select 18 bpp, and loosely18_en (DPI_COLOR_CODING) is not active, the number or pixels per line should be a multiple of four.
   - Configure dataen_active_low (DPI_CFG_POL): This bit configures the polarity of the dpidataen signal and sets it to be active low.
   - Configure vsync_active_low (DPI_CFG_POL): This bit configures the polarity of the dpivsync signal and sets it to be active low.
   - Configure hsync_active_low (DPI_CFG_POL): This bit configures the polarity of the dpihsync signal and sets it to be active low.
   - Configure shutd_active_low (DPI_CFG_POL): This bit configures the polarity of the dpishutdn signal and sets it to be active low.
   - Configure colorm_active_low (DPI_CFG_POL): This bit configures the polarity of the dpicolorm signal and sets it to be active low.

- Configure loosely18_en (DPI_COLOR_CODING): This bit configures if the pixel packing is done loosely or packed when dpi_color_coding is 18 bpp. This bit enables loosely packing.

3. Select the Video transmission mode to define how the processor requires the video line to be transported through the DSI link.
   - Configure the low-power transitions (VID_MODE_CFG): This defines the video periods which are permitted to go to low-power if there is time available to do so.
   - Configure frame_bta_ack_en (VID_MODE_CFG): This specifies if the controller should request the peripheral acknowledge message at the end of frames.
   - Configure lp_cmd_en (VID_MODE_CFG): This specifies that commands are to be transmitted in low-power.
   - Burst mode: For details about Burst mode, see "Video Transmission Modes" on page 60.
     - Configure the register field vid_mode_type (VID_MODE_CFG) with value 2'b1x.
     - Configure vid_pkt_size (VID_PKT_SIZE) with the size of the active line period, measured in pixels.
     - The fields vid_num_chunks and vid_null_size are ignored by the DWC_mipi_dsi_host.
   - Non-Burst mode: For details about the Non-Burst mode, see "Video Transmission Modes" on page 60.
     - Configure the vid_mode_type field (VID_MODE_CFG) with 2'b0x.
     - Configure the vid_mode_type field (VID_MODE_CFG) with 2'b00 to enable the transmission of sync pulses.
     - Configure the vid_mode_type field (VID_MODE_CFG) with 2'b01 to enable the transmission of sync events.
     - Configure the vid_pkt_size field (VID_MODE_CFG) with the number of pixels to be transmitted in a single packet.

       Selecting this value depends on the available memory of the attached peripheral, if the data is first stored, or on the memory you want to select for the FIFOs in DWC_mipi_dsi_host.
     - Configure the vid_num_chunks field (VID_NUM_CHUNKS) to the number of packets to be transmitted per video line. The value of vid_pkt_size * vid_num_chunks is the number of pixels per line of video, except if vid_num_chunks is 0, which disables the multi-packets. If you set it to 1, there is still only one packet per line, but it can be part of a chunk, followed by a null packet.
     - Configure the vid_null_size field (VID_NULL_SIZE) with the size of null packets to be inserted as part of the chunks. Setting it to 0 disables null packets.

4. Define the DPI Horizontal timing configuration as follows:
   - Configure the vid_hline_time field (VID_HLINE_TIME) with the time taken by a DPI video line measured in cycles of lane byte clock (for a clock lane at 500 MHz the lane byte clock period is 8 ns). When the periods of DPI clock and lane byte clock are not multiples, the value to program the vid_hline_time needs to be rounded. A timing mismatch is introduced between the lines due to the rounding of configuration values. If the DWC_mipi_dsi_host is configured not to go to low-power, this timing divergence accumulates on every line, introducing a significant amount of mismatch towards the end of the frame. The reason for this is that the DWC_mipi_dsi_host cannot re-synchronize on every new line because it transmits the blanking packets when the Horizontal Sync event occurs on the DPI interface. However, the accumulated mismatch should

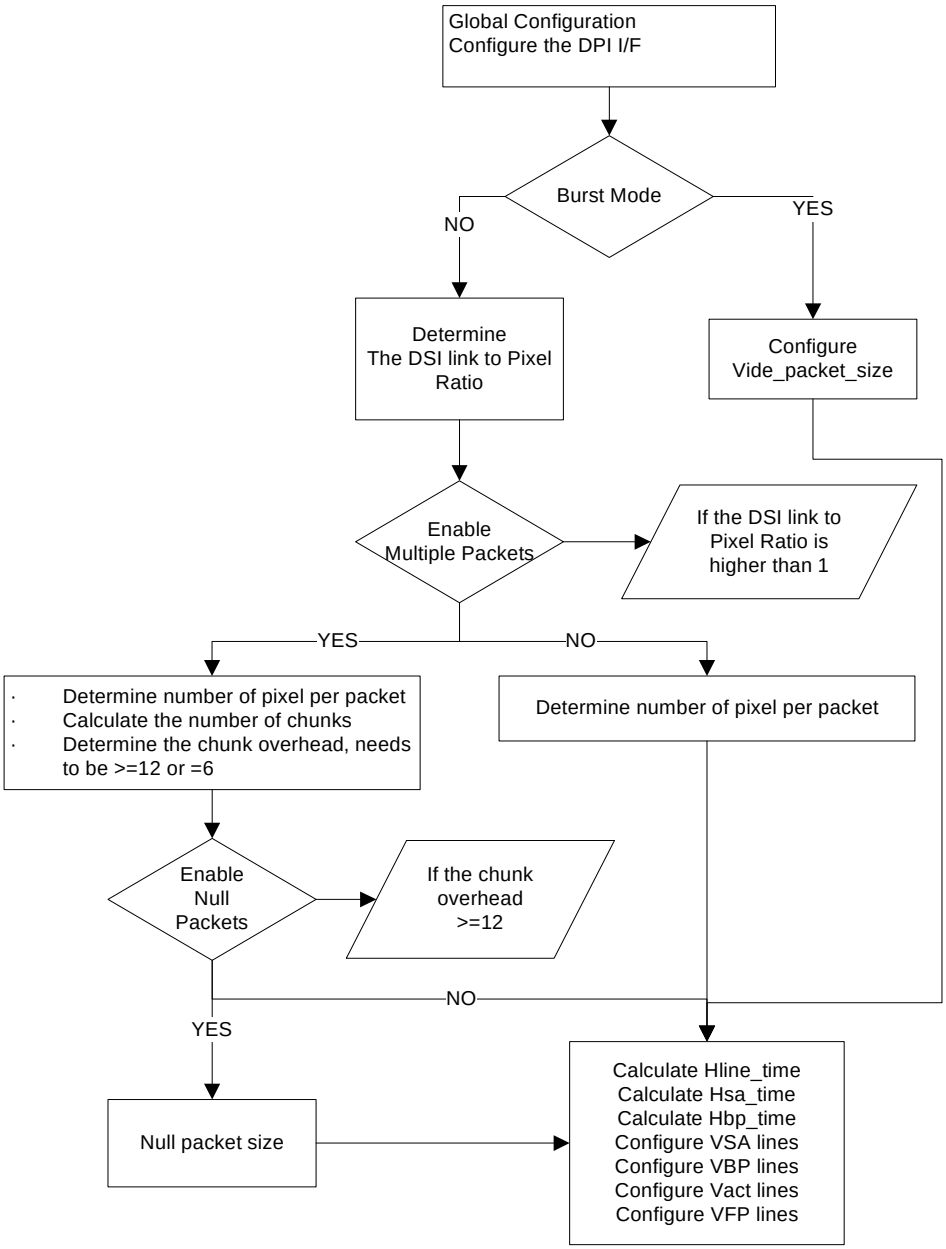
become extinct on the last line of a frame, where, according to the DSI specification, the link should always return to low-power regaining synchronization, when a new frame starts on a vertical sync event. If the accumulated timing mismatch is greater than the time in low-power on the last line, a malfunction occurs. This phenomenon can be avoided by configuring the DWC_mipi_dsi_host to go to low-power once per line.

- Configure the vid_hsa_time field (VID_HSA_TIME) with the time taken by a DPI Horizontal Sync Active period measured in cycles of lane byte clock (normally a period of 8 ns).
- Configure the vid_hbp_time field (VID_HBP_TIME) with the time taken by the DPI Horizontal Back Porch period measured in cycles of lane byte clock (normally a period of 8 ns). Special attention should be given to the calculation of this parameter.

5. Define the Vertical line configuration:
   - Configure the vsa_lines field (VID_VSA_LINES) with the number of lines existing in the DPI Vertical Sync Active period.
   - Configure the vbp_lines field (VID_VBP_LINES]) with the number of lines existing in the DPI Vertical Back Porch period.
   - Configure the vfp_lines field (VID_VFP_LINES) with the number of lines existing in the DPI Vertical Front Porch period.
   - Configure the v_active_lines field (VID_VACTIVE_LINES) with the number of lines existing in the DPI Vertical Active Period.

Figure 4-4 on page 57 illustrates the steps for configuring the DPI packet transmission.

Figure 4-4 DPI Packet Transmission Configuration Flow Diagram



Example 4-1 Configuration Example

The following is an example of DPI packet transmission configuration:

DPI video resolution:

- PCLK period = 50 ns
- HSA = 8 PCLK
- HBP = 8 PCLK
- HACT = 480 PCLK
- HFP = 24 PCLK
- VSA = 2 Line
- VBP = 2 Line
- VACT = 640 Line
- VFP = 4 Line

Configuration steps:

1. Global configuration

   n_lanes (PHY_IF_CFG)= 1, that is, two lanes available for HS transmission.

2. DPI interface configuration
   - dpi_vcid (DBI_VCID) = 2
   - dpi_color_coding (DPI_COLOR_CODING) =7, that is, 24 bpp
   - dataen_active_low (DPI_CFG_POL) = 1, that is, the signal is active low
   - vsync_active_low (DPI_CFG_POL) = 1, that is, the signal is active low
   - hsync_active_low (DPI_CFG_POL) = 1, that is, the signal is active low
   - shutd_active_low (DPI_CFG_POL) = 0, that is, the signal is active high
   - colorm_active_low (DPI_CFG_POL) = 0, that is, the signal is active high
   - loosely18_en (DPI_COLOR_CODING) is irrelevant since 18 bpp color mode is not selected
3. Video transmission mode configuration:
   a. Configure the low-power transitions:

      VID_MODE_CFG[13:8] = 6'b111111, that is, enable LP in all video period.

   b. Enable frame_bta_ack_en field (VID_MODE_CFG), that is, the DWC_mipi_dsi_host requests an acknowledge response message from the peripheral at the end of each frame.

   c. If you want to use the Burst mode, follow these steps:
      - vid_mode_type (VID_MODE_CFG) = 2'b1x
      - vid_pkt_size (VID_PKT_SIZE) = 480

d. If you want to use the non-Burst mode, follow these steps:

- For sync pulses, vid_mode_type (VID_MODE_CFG) = 2'b00 and for sync events, vid_mode_type (VID_MODE_CFG) = 2'b01
- If the bandwidth of DSI link is greater than the pixel bandwidth, you may want to activate multi-packets by setting vid_pkt_size (VID_PKT_SIZE) to a value greater than 1.

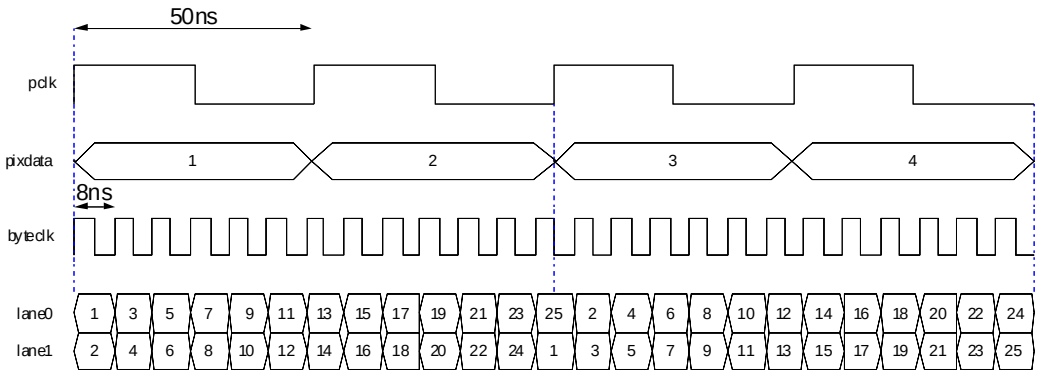
  Find an integer number of pixels that matches a period of bytes on the DSI link:

i. 1 pixel = 50 ns; (50/8)*number_of_lanes = 12.5

ii. 2 pixels = 100 ns; (100/8)*number_of_lanes = 25

iii. Configure vid_pkt_size (VID_PKT_SIZE) = 2 pixels

Figure 4-5 Byte Distribution of a Chunk



- num_chunks = HACT/vid_pkt_size = 240.
- Calculate overhead of a chunk:

  Bytes_per_pixel = 3 (DPI @ 24 bpp)

  Bytes_pixel_per_chunk = 2*3 = 6

  Bytes_of_video_packet_per_chunk = 4+6+2 = 12 (header + payload + CRC)

  Total_bytes_per_chunk = 25 (see Figure 4-5)

  Chunk_overhead = Total_bytes_per_chunk- Bytes_pixel_per_chunk = 19
- If Chunk_overhead >=12, configure vid_null_size. Otherwise, you would have to find a greater value to program vid_pkt_size with, to have less chunks and have room to make Chunk_overhead >= 6.
- Calculation of the Null packet payload size:

  Null_packet_overhead = 12

  vid_null_size (VID_NULL_SIZE) = Chunk_overhead - Null_packet_overhead = 7

4. Horizontal timing configuration:

- vid_hline_time (VID_HLINE_TIME) = (HSA+HBP+HACT+HFP)*(PCLK period/Clk Lane Byte Period)

  vid_hline_time = (8+8+480+24)*(50/8) = 3250
- vid_hsa_time (VID_HSA_TIME) = HSA*(PCLK period/Clk Lane Byte Period); hsa_time = 8*(50/8) = 50

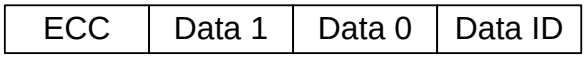
- vid_hbp_time (VID_HBP_TIME) = HBP*(PCLK period/Clk Lane Byte Period); hbp_time=8*(50/8) = 50

5. Vertical line configuration:
   - vid_vsa_lines (VID_VSA_LINES) = 2
   - vid_vbp_lines (VID_VBP_LINES) = 2
   - vid_vfp_lines (VID_VFP_LINES) = 4
   - v_active_lines (VID_VACTIVE_LINES) = 640

4.2.2 Video Mode with Stereoscopic Image Data

A short packet consists of the bytes shown in Figure 4-6.

Figure 4-6 Bytes in a Short Packet



When the SDF_3D register is written, the next VSS packet is sent with Data 0 byte set to 0x04, as the bit 3 indicates that the 3D control payload is present 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. This behavior takes place even if the new data coincides with the previous contents of the register. This way, each time it is changed, the new configuration is automatically sent. It is also possible to re-send the SDF_3D[5:0] upon request, when necessary, to meet any refresh requirements the peripheral might have. If the SDF_3D[16] bit is set, the 3D control configuration is automatically sent in every VSS packet.

The 3D control is also used to indicate that the 2D mode is on. When switching from a 3D mode to 2D, the next VSS packet contains data 0 set to 0x04 and data 1 has the two LSB set to '00' to indicate that the 3D and the 2D mode are not on.

4.2.3 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

4.2.3.1 Burst Mode

In this mode, the entire active pixel line is buffered into a FIFO and transmitted in a single packet with no interruptions. This transmission mode requires that the DPI Pixel FIFO has the capacity to store a full line of active pixel data inside it. This mode is optimally used if the difference between the pixel required bandwidth and DSI link bandwidth is very different. This enables the DWC_mipi_dsi_host to quickly dispatch the entire active video line in a single burst of data and then return to low-power mode.

4.2.3.2 Non-Burst Mode

In this mode, the processor uses the partitioning properties of the DWC_mipi_dsi_host to divide the video line transmission into several DSI packets. This is done to match the pixel required bandwidth with the DSI link bandwidth. With this mode, the controller configuration does not require a full line of pixel data to be stored inside the DPI Pixel FIFO. It requires only the content of one video packet.

4.2.3.3 Guidelines for Selecting the Burst or Non-Burst Mode

Selecting the Burst and Non-Burst mode is mainly dependent on the system configuration and the device requirements. Choose the video transmission mode that suits the application scenario. The Burst mode is more beneficial because it increases the probability of the link spending more time in the low-power mode, decreasing power consumption. However, the following conditions should be met for availing the maximum benefits from the Burst mode of operation:

- The DWC_mipi_dsi_host core should have sufficient pixel memory to store an entire pixel line to avoid the overflow of the internal FIFOs.
- The display device should support receiving a full pixel line in a single packet burst to avoid the overflow on the reception buffer.
- The DSI output bandwidth should be higher than the DPI system interface input bandwidth in a relation that enables the link to go to low-power once per line.

If the system cannot meet these requirements, it is likely that the pixel data will be lost causing the malfunctioning of the display device while using the Burst mode. These errors are related to the capabilities of the system to store the 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 requiring a full pixel line (lesser DPI RAM requirements in the DWC_mipi_dsi_host)
- Operation with devices that support only a small amount of pixel buffering (less than a full pixel line)

The DSI Non-Burst mode should be configured in such way that the DSI output pixel ratio matches with the DPI input pixel ratio, reducing the memory requirements on both host and/or device side. This is achieved by dividing a pixel line into several chunks of pixels and optionally interleaving them with null packets.

The following equations show how the DWC_mipi_dsi_host core transmission parameters should be programmed in Non-Burst mode to match the DSI link pixel output ratio (left hand side of the "=" sign) and DPI pixel input (right hand side of the "=" sign).

When the null packets are enabled:

Lanebyteclkperiod * vid_num_chunks (vid_pkt_size * bytes_per_pixel + 12 + vid_null_size) / number_of_lanes

= pixels_per_line * dpipclkperiod

When the null packets are disabled:

Lanebyteclkperiod * vid_num_chunks (vid_pkt_size * bytes_per_pixel + 6) / number_of_lanes

= pixels_per_line * dpipclkperiod

4.3 eDPI Interface

The eDPI interface is an enhanced DPI interface with an additional functional mode. This additional mode allows it to support the transmission of DCS memory write commands, reusing the pins and pixel color mapping, based on the MIPI DPI pin description. This interface has the following two modes of operation:

- Standard Video mode (DPI Standard Interface)
- Adapted Command mode (Synopsys Proprietary Interface)

To use the eDPI interface, select eDPI in the Select the system interface option in coreConsultant.

Note The eDPI interface does not operate concurrently with DBI and DPI interfaces.

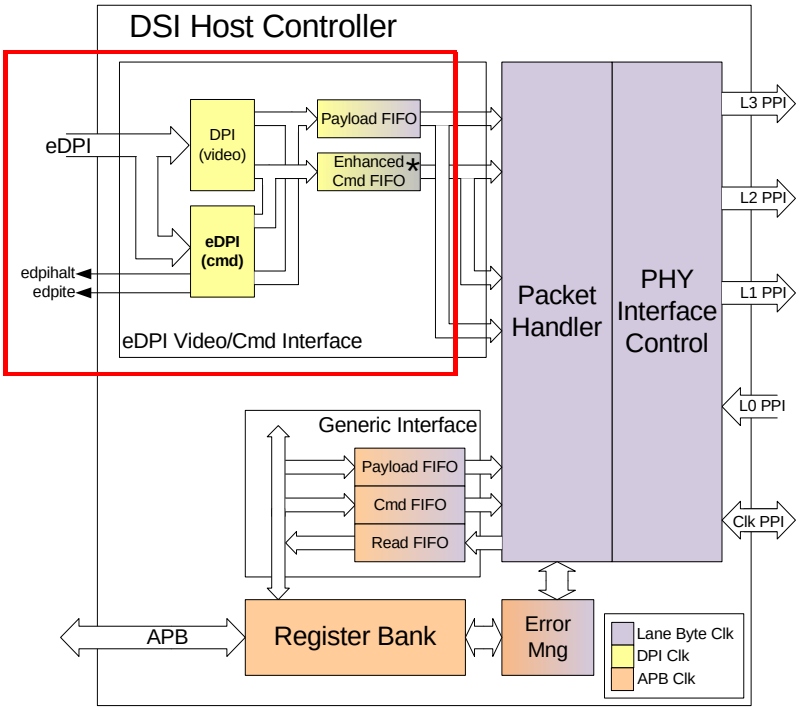
The standard Video mode of operation sets the interface to operate as a native DPI interface. For details about this mode of operation, see "DPI Interface" on page 51.

The eDPI interface, operating in Adapted Command mode, enables the system to input a stream of pixel information that is conveyed by DWC_mipi_dsi_host using the Command mode transmission (using the DCS packets). The eDPI interface also supports pixel input control rate signaling (edpihalt) and Tearing Effect report mechanism (edpite).

The Adapted Command mode allows sending large amounts of data through the memory_write_start (WMS) and memory_write_continue (WMC) DCS commands. It helps in delivering a wider bandwidth of data for the memory write operations sent in Command mode to MIPI displays and to refresh large areas of pixels in high resolution displays. If additional commands such as display configuration commands, read back commands, and tearing effect initialization are to be transferred, then the APB Slave Generic Interface should be used to complement the eDPI interface. If the eDPI configuration option is selected in the coreConsultant, the support for Generic packets is always selected in the configuration menu. This is required, because the eDPI interface can only support the WMS and WMC commands. The command mode display requires other commands for proper operation.

While in Adapted Command mode, the eDPI interface captures the data and control signals and conveys them to the FIFO interfaces that transmit them to the DSI link. It reuses the FIFOs of the DPI interface to store the payload data for the WMS and WMC commands as shown in Figure 4-7. Two different streams of data are presented at the interface: video control signals and pixel data. Depending on the interface color coding, the pixel data is disposed differently throughout the dpipixdata bus. Interface pixel color coding is presented in Figure 4-3 on page 53.

Figure 4-7 eDPI Block Diagram



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

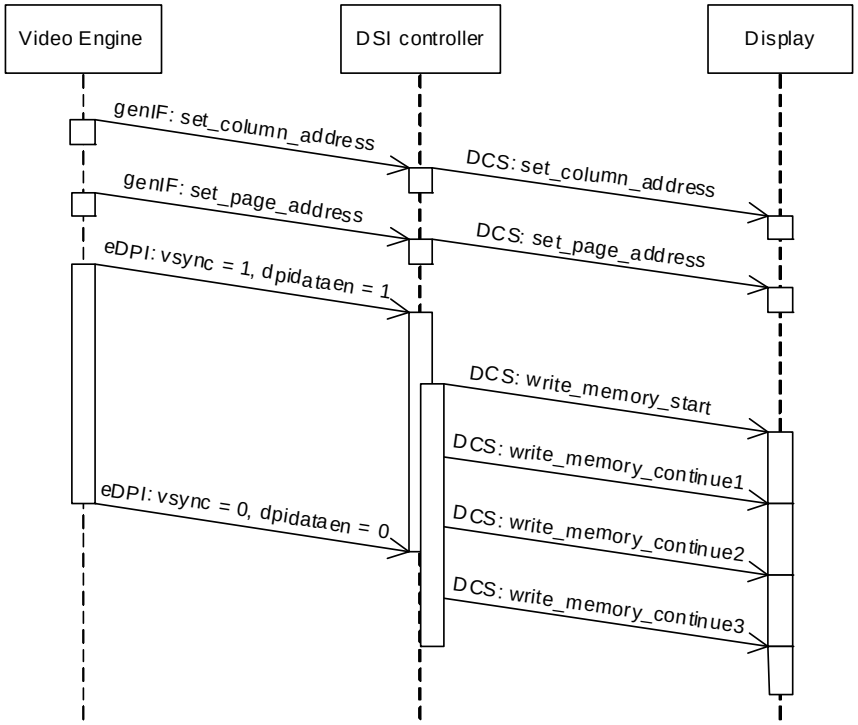
- Define the image area to be refreshed, by using the set_column_address and set_page_address DCS commands. The image area needs to be defined only once and remains effective until different values are defined.
- Define the pixel color coding to be used by using the dpi_color_coding field in the DPI_COLOR_CODING register. Also, define the Virtual Channel of the eDPI generated packets using the dpi_vcid field in the DBI_VCID register. These also need to be defined only once.
- Start transmitting the data through the eDPI interface.

To transmit the image data in Command mode using the eDPI interface, set cmd_video_mode bit of the MODE_CFG register to 1.

To transmit the image data in Video mode when using the eDPI interface, set cmd_video_mode bit of the MODE_CFG register to 0.

Figure 4-8 shows the eDPI usage flow.

Figure 4-8 eDPI Usage Flow



4.3.1 eDPI Interface in Adapted Command Mode

When the cmd_video_mode bit of the MODE_CFG register is set to 1, the eDPI pins assume the behavior corresponding to the Adapted Command mode.

In this mode, the host processor can use the eDPI interface to transmit a continuous stream of pixels to be written in the local frame buffer of the peripheral. It uses a pixel input bus to receive the pixels and a control signal, dpivsync_edpiwms, 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 that is input through the interface. When this pixel counter reaches edpi_allowed_cmd_size, a command is written into the command FIFO and the packet is ready to be transmitted through the DSI link.

If dpivsync_edpiwms falls before the counter reaches the value of shadowed edpi_allowed_cmd_size, a WMS command is issued to the command FIFO with Word Count (WC) set to the amount of bytes that correspond to the value of the counter. If more than edpi_allowed_cmd_size number of pixels are received (shadowed value), a WMS command is sent to the command FIFO with WC set to the number of bytes that correspond to edpi_allowed_cmd_size and the counter is restarted.

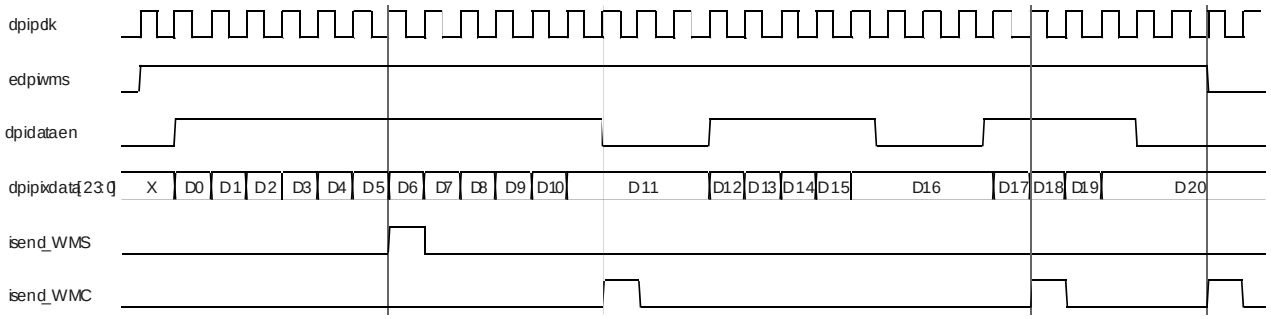
Note The value of edpi_allowed_cmd_size is defined in pixels and not in bytes as the WC packet field. The DWC_mipi_dsi_host automatically converts the pixels to bytes.

After the first WMS command has been written to the FIFO, the circuit behaves in a similar way, but issues WMC commands instead of WMS commands. The process is repeated until dpivsync_edpiwms falls. The

core automatically starts sending a new packet when dpivsync_edpiwms falls or edpi_allowed_cmd_size limit is reached.

The payload FIFOs always operate according to dpidataen signal. This means that the FIFOs may be written in chunks if necessary. Figure 4-9 shows an example in which the value of edpi_allowed_cmd_size corresponds to six words of 24 bits per pixel. After dpivsync_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 dpivsync_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 4-9 WMS and WMC Commands in Automatic Partitioning Mode



4.3.2 Support for Tearing Effect

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 process of reading the content of its frame buffer. Some displays have a separate pin for the same function, which is not covered in the DSI specification. According to the DSI specification, to use the tearing effect functionality, a set_tear_on DCS command should be issued through the APB interface using the Generic interface registers.

The DWC_mipi_dsi_host performs a double Bus Turn-Around (BTA) after sending the set_tear_on command 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 DWC_mipi_dsi_host by a D-PHY trigger event. The DWC_mipi_dsi_host then decodes the trigger and indicates the event by pulsing edpite signal during one dpipclk cycle.

To use this function, it is necessary to issue a set_tear_on command after the update of the display 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 does not automatically generate the tearing effect request (double BTA) after a WMS/WMC sequence for flexibility purposes. 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 command in the DWC_mipi_dsi_host implementation.

Configure the following registers to activate the tearing effect:

- CMD_MODE_CFG: tear_fx_en
- PCKHDL_CFG: bta_en

4.3.3 Halt Functionality in Adapted Command Mode

The edpihalt output signal rises each time either the payload or the command FIFO is about to become full after two clock cycles, and falls in the opposite condition. While the edpihalt is asserted, dpivsync_edpiwms can go low and high, signaling the end and the start of a new command. However, if the dpivsync_edpiwms goes low and high during edpihalt, dpidataen cannot be asserted after the rising edge of dpivsync_edpiwms, until edpihalt is de-asserted. In this case (edpihalt is asserted), the two extra dpidataen pulses can only occur before dpivsync_edpiwms goes low.

Figure 4-10 shows the waveform for a sample related to edpihalt. In this case, the system has two additional clock cycles of latency to react to edpihalt.

Figure 4-10 Halt Functionality for a System Delay of two Clock Cycles

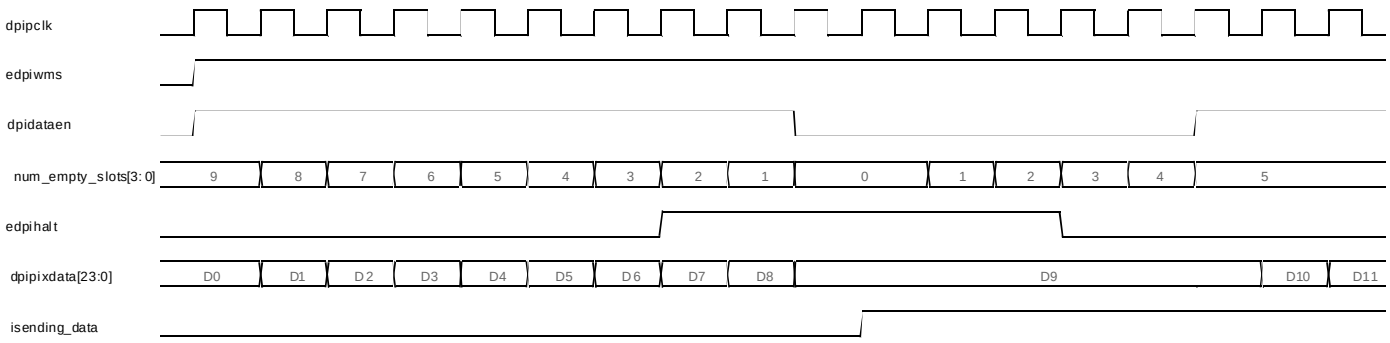


Figure 4-11 shows some valid cases related to edpihalt and dpidataen with edpihalt asserted.

Figure 4-11 Valid Cases of Halt Functionality and dpidataen

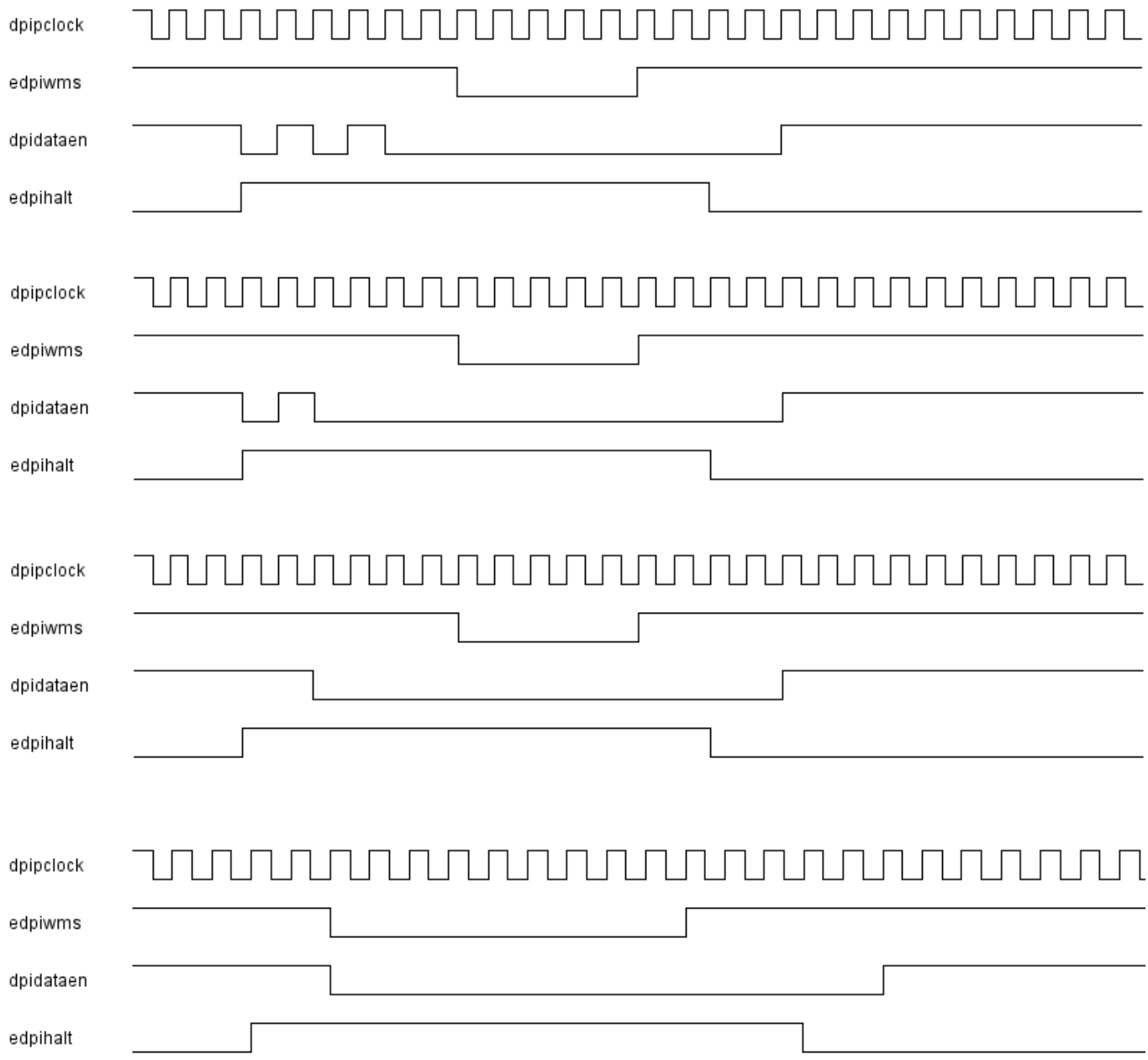


Figure 4-12 shows some invalid cases related to edpihalt and dpidataen with edpihalt asserted.

Figure 4-12 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 dpivsync_edpiwms rises and remains valid until the next time dpivsync_edpiwms 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, if the value of the edpi_allowed_cmd_size field is set to a very small value, edpihalt may rise soon when three commands are sent to this FIFO, even if the

payload FIFO is close to empty. For that reason, utilization of the core can be optimized by choosing 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, dpipclk should remain active at all times.

Figure 4-13 shows the eDPI usage when the payload is received in chunks.

Figure 4-13 Flow when Payload is Received in Chunks

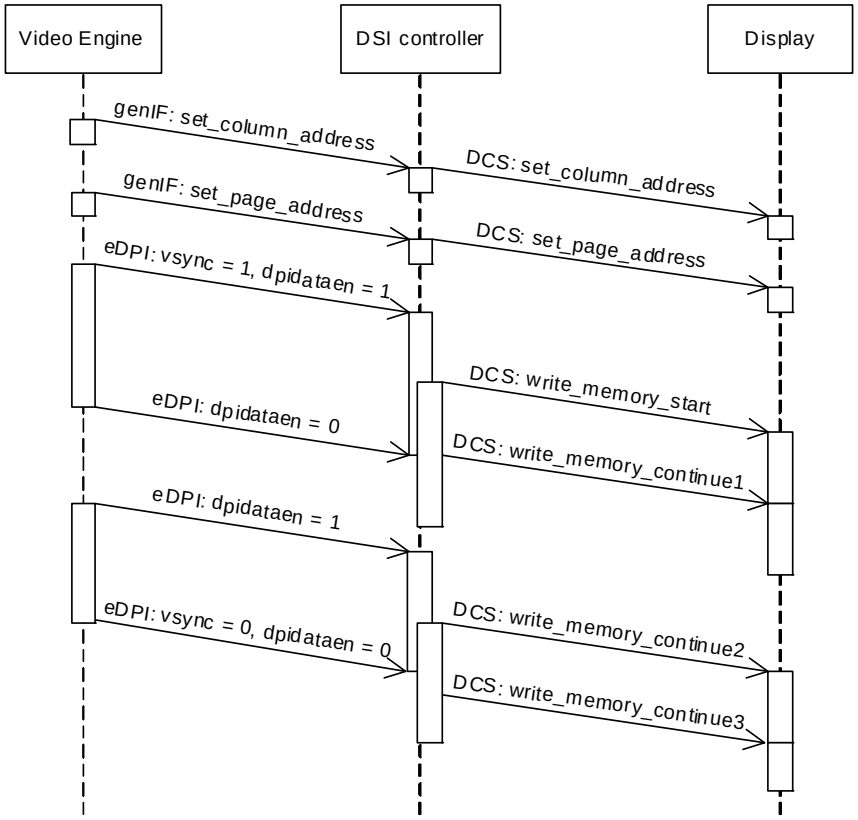


Figure 4-14 shows the eDPI usage when the video engine is faster than the DSI Link.

Figure 4-14 Flow of eDPI when Video Engine Faster than the DSI Link

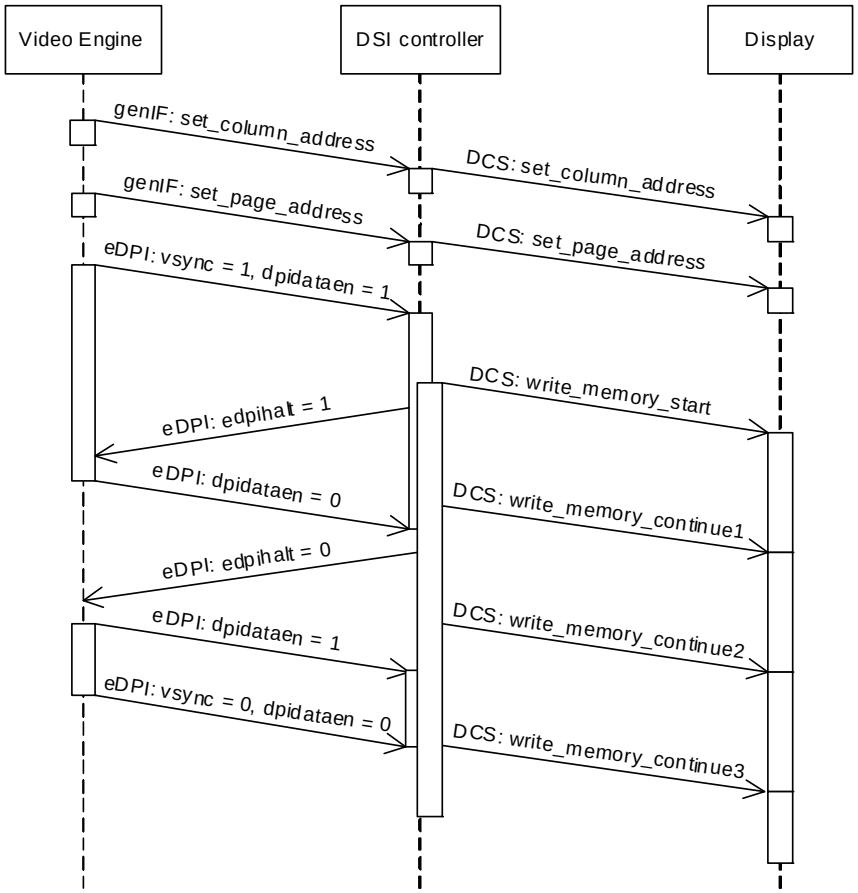
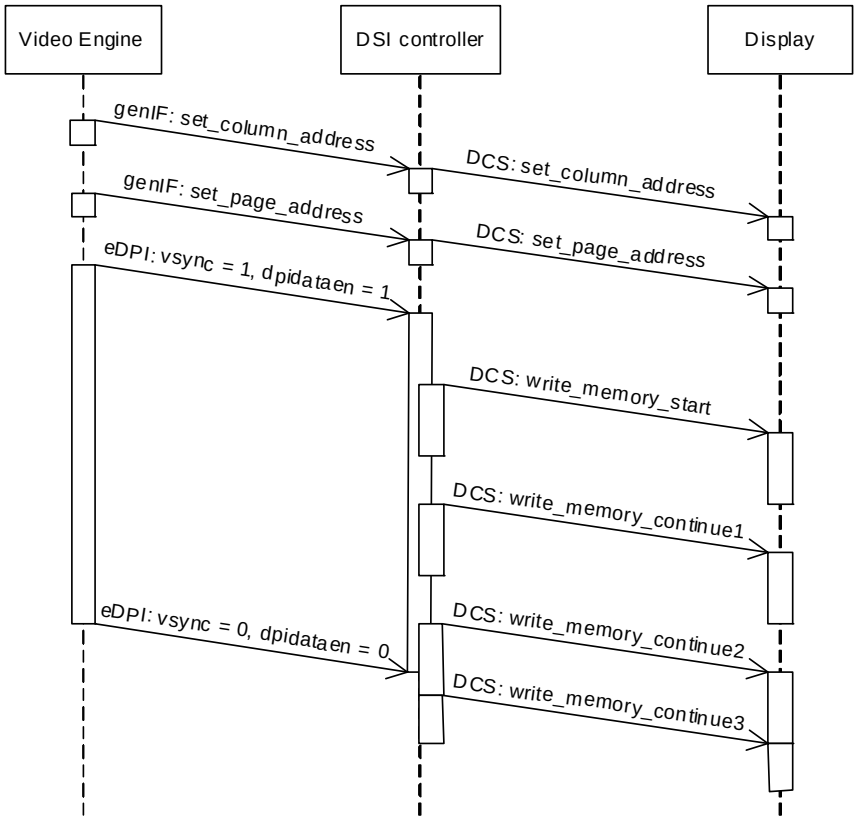


Figure 4-15 shows the eDPI usage when the video engine is slower than the DSI Link.

Figure 4-15 Flow when Video Engine is Slower than the DSI Link



4.4 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 are not constrained to comply with the DCS specification, and can include generic commands described in the DSI specification as manufacturer-specific.

To use the APB Slave Generic interface, select the Enable the support for Generic Packets option in coreConsultant.

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

The DWC_mipi_dsi_host 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. The GEN_PLD_DATA register has two distinct functions based on the operation. Writing to 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 to this

register triggers the transmission of the packet implying that for a long Command mode packet, the packet's payload needs to be written in advance in 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.

Generic interface packets are always transported using one of the DSI transmission modes; Video mode or Command mode. If neither of these modes is selected, the packets are not transmitted through the link and the related FIFOs eventually get overflowed.

If the system does not require Generic interface support, this functionality can be disabled in the coreConsultant GUI, and the interfaces with the 2-Port RAMs that buffer command, payload, and read data are removed.

The selection of the Generic interface is highly recommended to ensure interoperability with most of the DSI displays. The DPI and DBI interfaces do not support vendor-specific commands that are generally required to perform display initialization. If the eDPI interface is selected, Generic interface is automatically selected.

4.4.1 Packet Transmission Using the Generic Interface

The transfer of packets 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.

- The DSI link pixel bit rate when using solely APB is equal to (APB clock frequency) * 16 Mbps.
- When using only the APB interface, the theoretical DSI link maximum bit rate is given by the following formula:

  DSI link maximum bit rate = APB clock frequency (in MHz)* 32 / 2 Mbps.

  In the formula, 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 bandwidth is dependent on the APB clock frequency; the available bandwidth 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 should operate in the Command mode only and the APB interface should be the only data source that is 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. When the payload data is for the command parameters, place the first byte to be transmitted in the least significant byte position of the APB data bus. For more information, refer to "GEN_PLD_DATA" on page 155.

After writing the payload, write the packet header into the command FIFO. For more information about the packet header organization on the 32-bit APB data bus, so that it is correctly stored inside the Command FIFO, refer to "GEN_HDR" on page 155.

When the payload data is for a memory write command, it contains pixel information and it should follow the pixel to byte conversion organization referred in the Annexure A of the DCS specification. Figure 4-16 to Figure 4-20 show how the pixel data should be organized in the APB data write bus. The memory write commands are conveyed in DCS long packets. DCS long packets are encapsulated in a DSI packet. The DSI specifies that the DCS command should be present in the first payload byte of the packet. This is also included in the diagrams. In Figure 4-16 to Figure 4-20, the Write Memory Command can be replaced by the DCS command Write Memory Start and Write Memory Continue.

Figure 4-16 24 bpp APB Pixel to Byte Organization

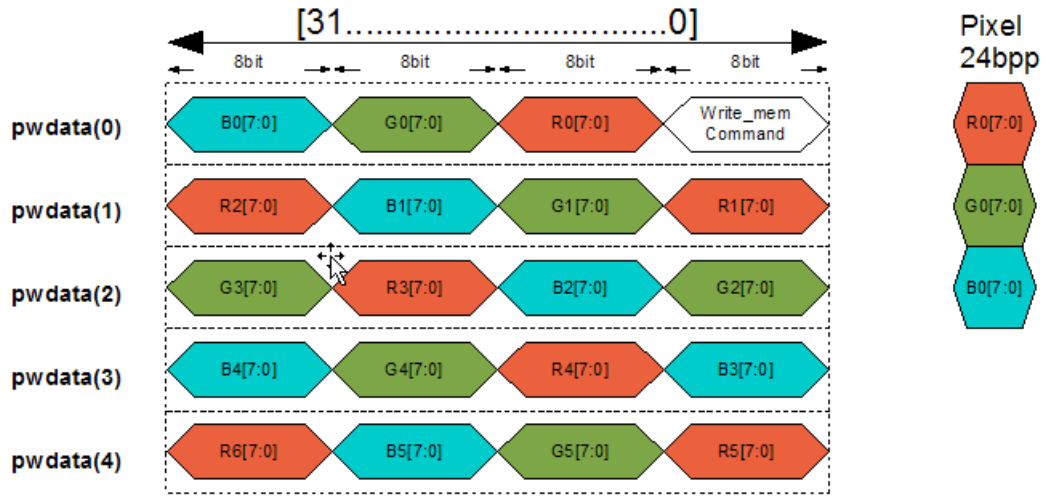


Figure 4-17 18 bpp APB Pixel to Byte Organization

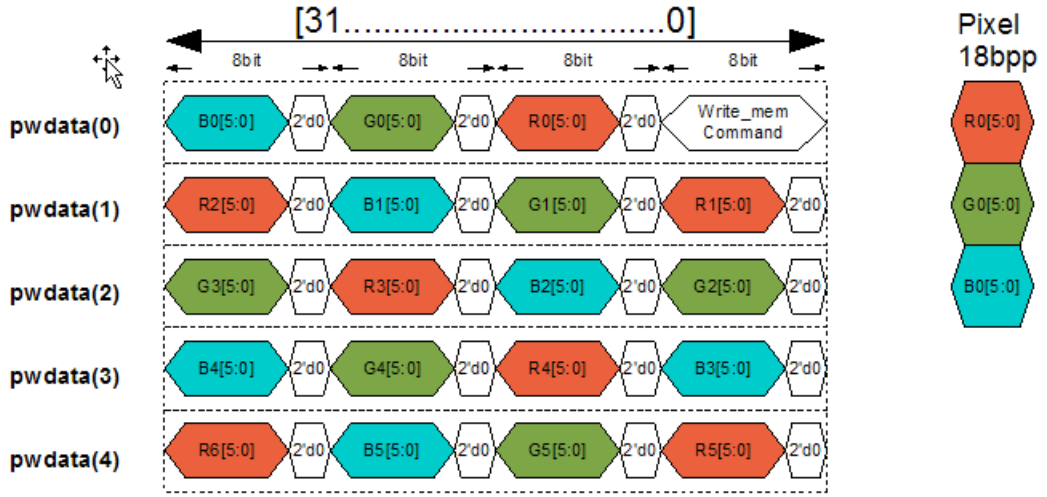


Figure 4-18 16 bpp APB Pixel to Byte Organization

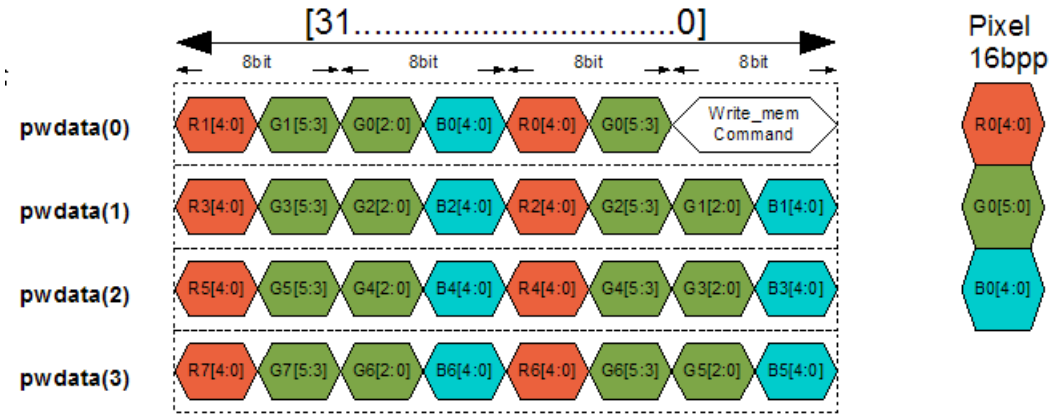


Figure 4-19 12 bpp APB Pixel to Byte Organization

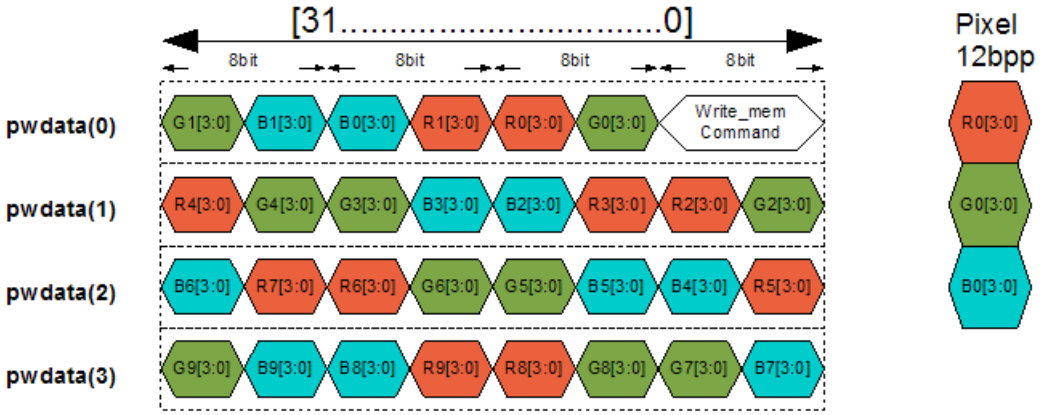
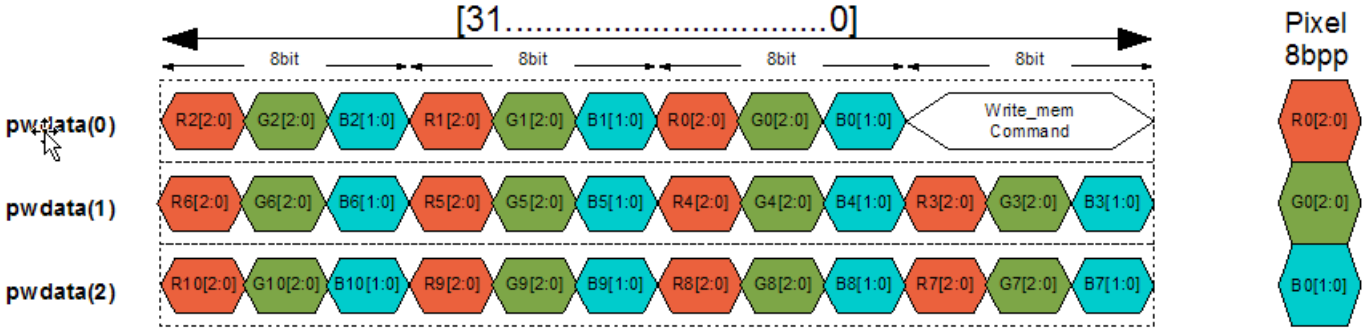


Figure 4-20 8 bpp APB Pixel to Byte Organization



4.5 Selection of Appropriate Interface

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

DBI

The DBI Interface is used with displays that support the Command mode for the DCS commands. The Command mode display device contains its own video timing control generation and a frame buffer. The DWC_mipi_dsi_host is required to fill the frame buffer of the device with pixel write commands and the device displays the content of the frame buffer on the display device. The Command mode display device also contains several commands to control its operation. These commands can be either read or write commands, and therefore, a bidirectional link is required.

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

DPI

The DPI Interface is used with displays that support the Video mode. The Video mode display device requires a continuous stream of pixel data and synchronism signals to be provided through the DSI link with accurate timing precision. The Video mode display has very low control capability. It can support only two commands - ShutDown and ColorMode and does not support the read commands. Therefore, a unidirectional link can be defined for Video mode display, unless any other interface in the system requires a bidirectional link.

Most of the Video mode displays support very high resolutions and have high bandwidth requirements from the DSI link. For example, Full HD 1080p24Hz may require a bandwidth of approximately 3.3 Gbps. The DWC_mipi_dsi_host can support up to 4 Gbps bandwidth, assuming that each data lane is operating at 1 Gbps and there are four data lanes. Therefore, the DPI interface can be selected if the display supports the Video mode and requires a pixel data bandwidth up to 4 Gbps for display refreshment.

eDPI

The eDPI interface supports the DCS memory write commands with the same bandwidth rates as the DPI interface. It is an ideal interface for the Command mode displays that have high bandwidth requirements. The eDPI interface supports only 16/18/24 bpp color depth modes. Therefore, the eDPI interface should be selected if the Command mode display requires a pixel data bandwidths of more than 660 Mbps for display refreshment, and is confined to of 16, 18, or 24 bpp modes.

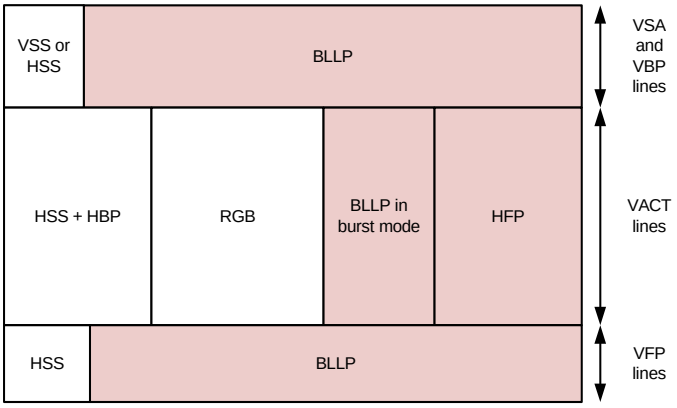
Several host systems require the capability to support a broader range of displays, both Video and Command modes. The eDPI interface is an efficient interface for these systems, given the particularity of the eDPI operational modes, Standard and Advanced, that can address both Video and Command mode displays respectively, sharing the same resources and interface pins of a native DPI interface.

4.6 Transmission of Commands

4.6.1 Transmission of Commands in Video Mode

The DSI controller supports the transmission of commands, both in high-speed and low-power, while in Video mode. The DSI 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 4-21.

Figure 4-21 Command Transmission Periods within the Image Area

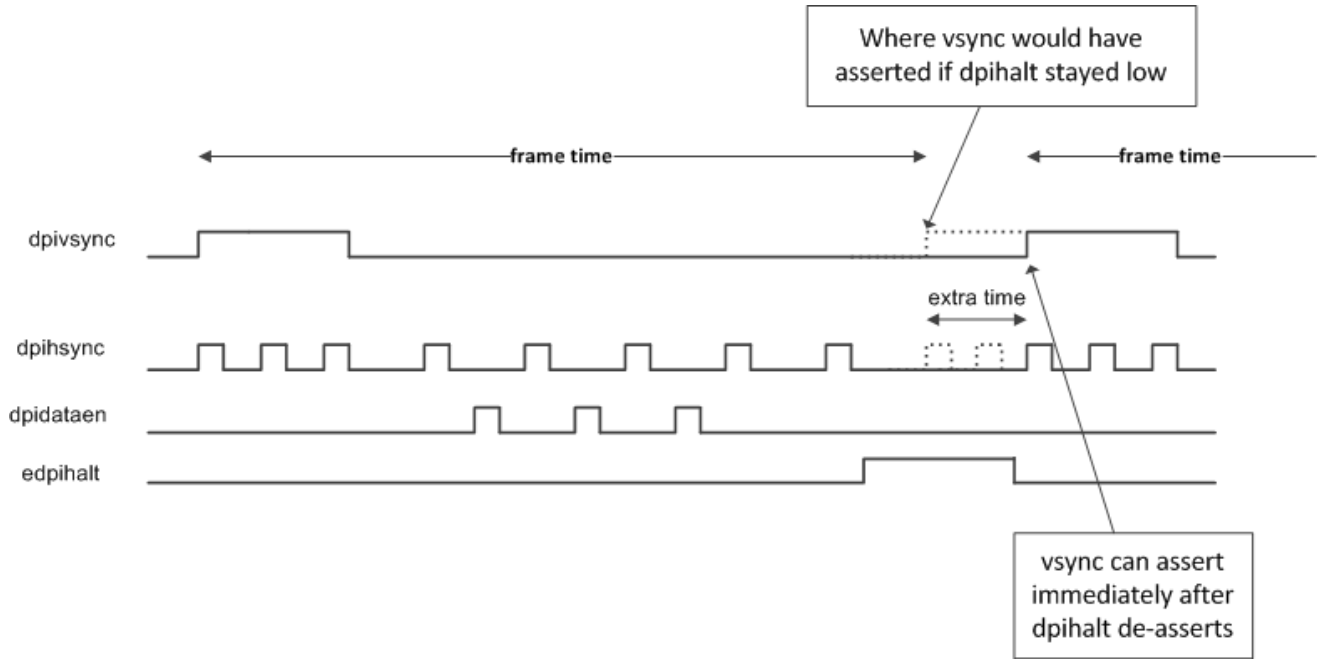


Commands are transmitted in the blanking periods after the following packets/states:

- 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, large commands can also be sent during the last line of a frame. In that case, the line time for the Video mode is violated and the edpihalt signal is set to request the DPI video timing signals to remain inactive. Only 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 4-22.

Figure 4-22 Transmission of Commands on the Last Line of a Frame



Only one command is transmitted per line, even in the case of the last line of a frame but one command is possible for each line.

There can be only one command sent in low-power per line. However, one low-power command is possible for each line. In high-speed, the DWC_mipi_dsi_host can send more than one command, as many as it determines to fit in the available time.

The DWC_mipi_dsi_host avoids sending commands in the last line because it is possible that the last line is shorter than the other ones. For instance, the line time (tL) could be half a cycle longer than the tL on the DPI interface, that is, each line in the frame taking half a cycle from time for the last line. This results in the last line being (½ cycle) x (number of lines -1) shorter than tL.

If a command is being transmitted, the edpihalt signal asserts in the last line. The dpivsync_edpiwms signal can assert immediately after edpihalt de-asserts.

The dpicolorm and dpishutdn input signals are also able to trigger 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 are not sent in the VACT region. If the lp_cmd_en bit of the VID_MODE_CFG register is 1, these commands are sent in LP mode. In LP mode, the outvact_lpcmd_time field of the DPI_LP_CMD_TIM register is used to determine 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 does not transmit these commands on the last line.

If the frame_bta_ack_en field is set in the VID_MODE_CFG register, a BTA is generated by DWC_mipi_dsi_host after the last line of a frame. This may coincide with a write command or a read

command. In either case, the edpihalt signal is held asserted until an acknowledge is received (control of the DSI bus is returned to the host).

4.6.2 Transmission of Commands in Low-Power

DWC_mipi_dsi_host can be configured to send the LP commands during the HS 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 LP mode to Horizontal Front Porch (HFP), Vertical Sync Active (VSA), Vertical Back Porch (VBP), and Vertical Front Porch (VFP) regions.

4.6.2.1 Calculating the Time to Transmit Commands in LP 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 LP 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 4-23 illustrates the timing intervals for the Video mode in non-Burst with sync pulses and Figure 4-24 illustrates the timing intervals for the Video mode in Burst and non-Burst with sync events.

Figure 4-23 outvact_lpcmd_time for Non-Burst with Sync Pulses

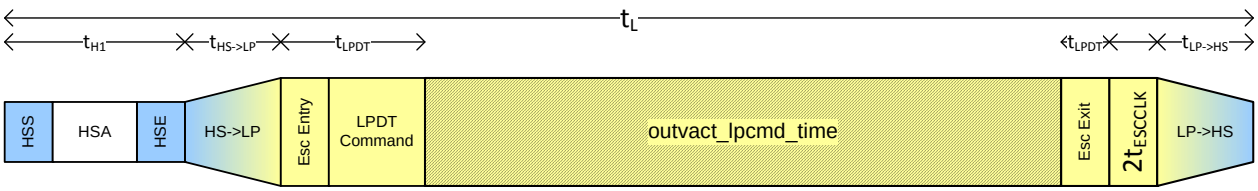
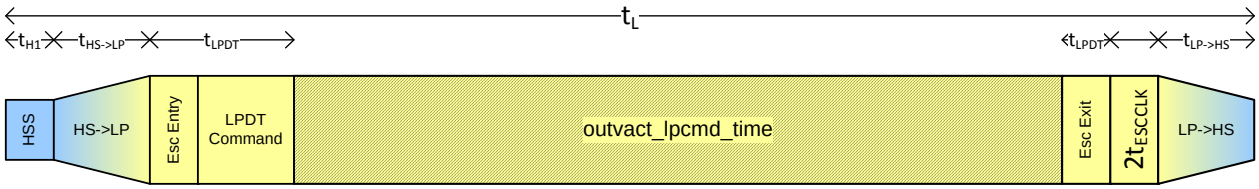


Figure 4-24 outvact_lpcmd_time for Burst or Non-Burst with Sync Events



This time is calculated as follows:

$$outvact\_lpcmd\_time = (t_L - (t_{H1} + t_{HS->LP} + t_{LP->HS} + t_{LPDT} + 2t_{ESCCLK}))/(2 \times 8 \times t_{ESCCLK})$$

Where

- t_L = Line time
- t_{H1} = Time of the HSA pulse for sync pulses mode (Figure 4-23) or the time to send the HSS packet, including EoTp (Figure 4-24)
- $t_{HS->LP}$ = Time to enter the LP mode
- $t_{LP->HS}$ = Time to leave the LP mode

- t_{LPDT} = D-PHY timing related with Escape Mode Entry, LPDT Command, and Escape Exit. According to the D-PHY specification, this value is always 11 bits in LP (or 22 TX Escape clock cycles).
- t_{ESCCLK} = Escape clock period as programmed in the tx_esc_clk_division field of the CLKMGR_CFG register
- $2xt_{ESCCLK}$ = Delay imposed by the core 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. The maximum size of a command that can be transmitted in LP mode is limited to 255 bytes by this field. You must program this register to a value greater than or equal to 4 bytes for the transmission of the DCTRL commands, such as shutdown and colorm in LP mode.

Consider an example of a frame with 12.4 µs 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 µs * 20 MHz/2). Still, you need to take into consideration the D-PHY protocol and PHY timings. The following are the assumptions:

- 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
- D-PHY takes 180 ns to transit from LP to HS modes (PHY_TMR_CFG: phy_lp2hs_time = 18)
- D-PHY takes 200 ns to transit from HS to LP modes (PHY_TMR_CFG: phy_hs2lp_time = 20)
- t_{HSA} = 420 ns

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

outvact_lpcmd_time= (12.4 µs - (420 ns + 180 ns +200 ns +(22×50 ns + 2×50 ns)))/(2×8×50 ns) = 13 bytes

4.6.2.2 Calculating the Time to Transmit the Commands in LP 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 LP 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 4-25 shows the timing intervals for video mode in Non-Burst with sync pulses. Figure 4-26 shows the timing intervals for video mode in non-Burst with sync events. Figure 4-27 shows the Burst video mode.

Figure 4-25 invact_lpcmd_time for Non-Burst with Sync Pulses

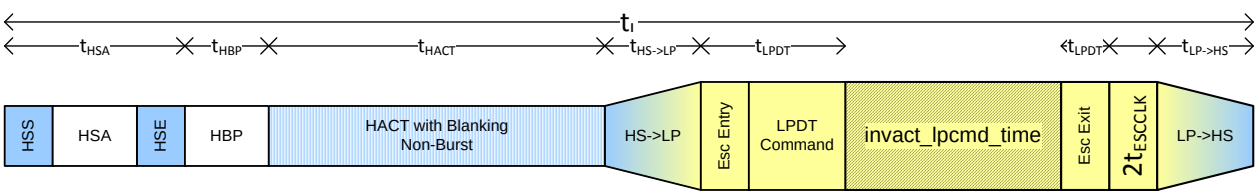


Figure 4-26 invact_lpcmd_time for Non-Burst with Sync Events

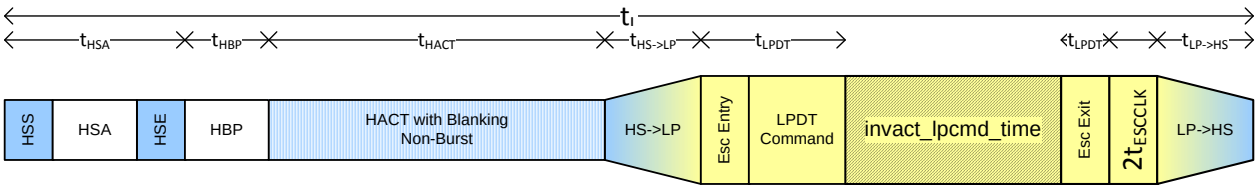
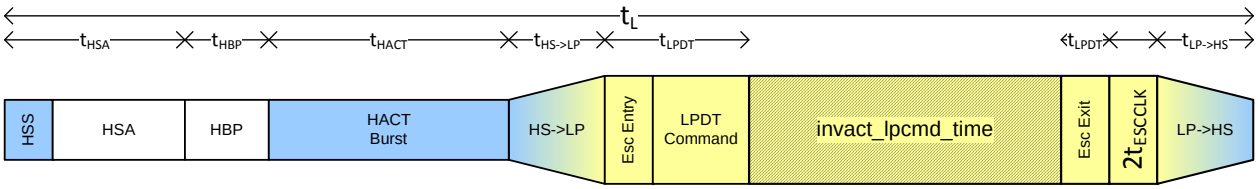


Figure 4-27 invact_lpcmd_time for Burst Mode



This time is calculated as follows:

$invact\_lpcmd\_time = (t_L - (t_{HSA}+t_{HBP}+t_{HACT}+ t_{HS->LP} + t_{LP->HS} + t_{LPDT} + 2t_{ESCCLK}))/(2\times8\times t_{ESCCLK})$

Where

- t_L = Line time
- t_{HSA} = Time of the HSA pulse (VID_HSA_TIME)
- t_{HBP} = Time of Horizontal back porch (VID_HBP_TIME)
- t_{HACT} = Time of Video active. For Burst mode, the Video active is time compressed and is calculated as follows:

  t_{HACT} = vid_pkt_size * Bytes_per_Pixel /Number_Lanes * $t_{Lane\_byte\_clk}$
- t_{ESCCLK} = 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.

Consider an example of a frame with 16.4 µs 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, 420 bits = 16,4 µs * 20 Mhz/2). Still, since it is the Vertical Active region of the frame, take into consideration the HSA, HBP, and HACT timings apart from the D-PHY protocol and PHY timings. The following is assumed:

- Lane byte clock period is 10 ns (800 Mbits per Lane)
- Escape clock period is 50 ns (CLKMGR_CFG: tx_esc_clk_division = 5)
- D-PHY takes 180 ns to transit from LP to HS modes (PHY_TMR_CFG: phy_lp2hs_time = 18)
- D-PHY takes 200 ns to transit from HS to LP modes (PHY_TMR_CFG: phy_hs2lp_time = 20)
- t_{HSA} = 420 ns

- t_{HBP} = 800 ns
- t_{HACT} = 12800 ns to send 1280 pixel at 24 bpp
- Video is transmitted in non-Burst mode
- DWC_mipi_dsi_host is configured for four lanes

In this example, consider that you send video in Non-burst mode. The invact_lpcmd_time is calculated as follows:

invact_lpcmd_time = (12.4 µs -(420 ns + 800 ns + 8.8 µs + 180 ns +200 ns +(22×50 ns + 2×50 ns))/(2×8×50 ns) = 1 bytes

Only one byte can be transmitted 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.

invact_lpcmd_time
= (12.4 µs - (420 ns+800 ns + (1240×3/4×10 ns) + 180 ns + 200 ns + (22×50 ns + 2×50 ns)/(2×8×50 ns)
= 5 bytes

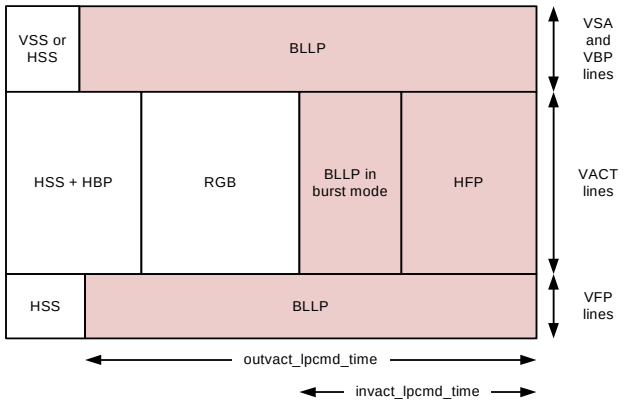
For Burst mode, the invact_lpcmd_time is 5 bytes and you can effectively send a 4-byte short packet.

4.6.2.3 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 4-28 illustrates the meaning of invact_lpcmd_time and outvact_lpcmd_time, matching them with the shaded areas and the VACT region.

Figure 4-28 Location of outvact_lpcmd_time and invact_lpcmd_time in the Image Area



4.6.3 Transmission of Commands in High-Speed

If the lp_cmd_en bit of the VID_MODE_CFG register is 0, the commands are sent in high-speed in Video Mode. In this case, the DWC_mipi_dsi_host automatically determines the area where each command can be sent and no programming or calculation is required.

4.6.4 Read Command Transmission

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

max_rd_time = Time to transmit the read command in LP mode + Time to enter and leave LP mode + 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. The max_rd_time field is used in both HS and LP mode to determine 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:

max_rd_time = phy_hs2lp_time + Time to return the read data packet from the peripheral device + phy_lp2hs_time

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

max_rd_time = phy_hs2lp_time + LPDT command time + Read command time in LP mode + Time to return the data read from the peripheral device + phy_lp2hs_time

Where,

- LPDT command time = (8 * Host escape clock period) / Lane byte clock period
- Read command time in LP mode = (32 * host escape clock period) / lane byte clock period

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. Ensure that max_rd_time x Lane byte clock period is less than outvact_lpcmd_time x 8 x Escape clock period of the host, otherwise, the read commands are dispatched on the last line of a frame. If it is necessary to read a large number of parameters (> 16), increase the max_rd_time while the read command is being executed. When the read has completed, decrease the max_rd_time to a lower value.

If a read command is issued on the last line of a frame, the edpihalt signal gets asserted and stays asserted until the read command is in progress. The video transmission should be stopped during this period.

4.6.5 Clock Lane in Low-Power Mode

To reduce the power consumption of the D-PHY, the DWC_mipi_dsi_host, when not transmitting in the high-speed mode, allows the clock lane to enter into the low-power mode. The controller automatically handles the transition of the clock lane from HS (Clock lane active sending clock) to LP state without direct intervention by the software. This feature can be enabled by configuring the phy_txrequestclkhs and the auto_clklane_ctrl bits of the LPCLK_CTRL register.

In the Command mode, the DWC_mipi_dsi_host can place the clock lane in the low-power mode when it does not have any HS packets to transmit.

In the Video mode (DPI interface), 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 the low-power mode and not compromise the video data transmission of pixel data and sync events.

Along with a correct configuration of the Video mode (see "DPI Interface" on page 51), the DWC_mipi_dsi_host 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 HS to LP in clock lane / Byte clock period in HS (lanebyteclk)
- phy_clklp2hs = Time from LP to HS in clock lane / Byte clock period in HS (lanebyteclk)

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

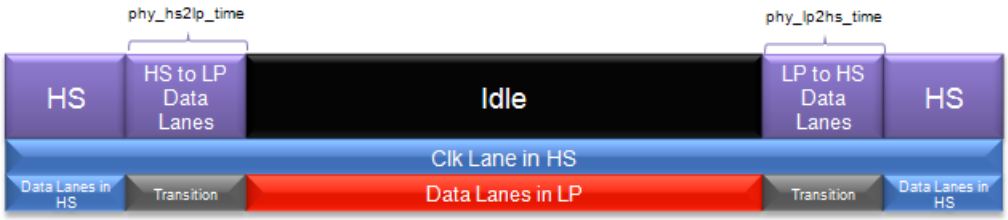
The DWC_mipi_dsi_host decides the best approach to follow regarding power saving out of the following three possible scenarios:

- There is no sufficient time to go to the low-power mode. Therefore, blanking period is added as shown in Figure 4-29.

Figure 4-29 Clock Lane and Data Lanes in HS

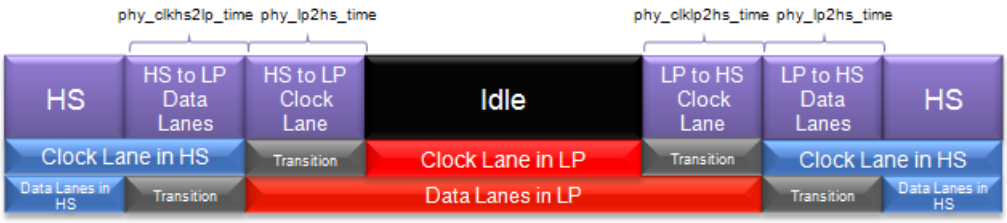
- There is sufficient time for the data lanes to go to the low-power mode but not enough time for the clock lane to enter the low-power mode.

Figure 4-30 Clock Lane in HS and Data Lanes in LP



- There is sufficient time for both data lanes and the clock lane to go to the low-power mode.

Figure 4-31 Clock Lane and Data Lanes in LP

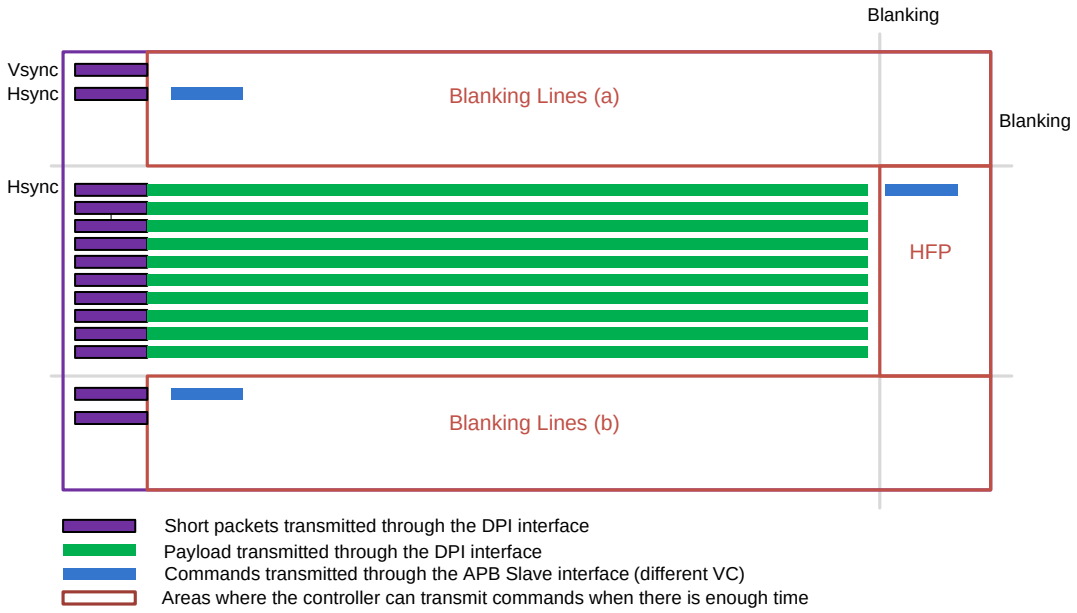


4.7 Virtual Channels

The DWC_mipi_dsi_host supports choosing the Virtual Channel (VC) for use for each interface. Using multiple Virtual Channels, the system can address multiples displays at the same time, when each display has a different Virtual Channel identifier.

When the DPI interface is configured for a particular Virtual Channel, it is possible to use the APB Slave Generic interface to issue the commands while the video stream is being transmitted. With 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 transmission of sideband packets such as the ones from the Generic Interface are only transported when there is time available within the video stream transmission. The DWC_mipi_dsi_host identifies the available time periods and uses them to transport the Generic interface packets. Figure 4-32 illustrates where the DWC_mipi_dsi_host inserts the packets from the APB Generic interface within the video stream transmitted by the DPI interface.

Figure 4-32 Command Transmission by the Generic Interface



When the DBI interface is configured 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 also addresses the multiple displays with different Virtual Channels. Because in Command mode there are no particular

timing requirements as opposed to Video mode, the priority is given to the DBI generated packets and only then the Generic interface packets are considered. This means that as long as there are packets from the DBI interface 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. Because the Generic Interface is not restricted to any particular Virtual Channel through configuration, it is possible to issue the packets with different Virtual Channels. This enables the interface to time multiplex the packets to be provided to the displays with different Virtual Channels.

You can 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_vcid field configures the DPI Virtual Channel ID that is indexed to the Video mode packets.
- DBI_VCID: The dbi_vcid field configures the DBI Virtual Channel ID that is 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.

5 Signals

This chapter helps you to understand the DWC_mipi_dsi_host signals and their properties. It describes the naming conventions, I/O mapping, width, dependency, and the behavior with various interfaces.

This chapter has the following sections:

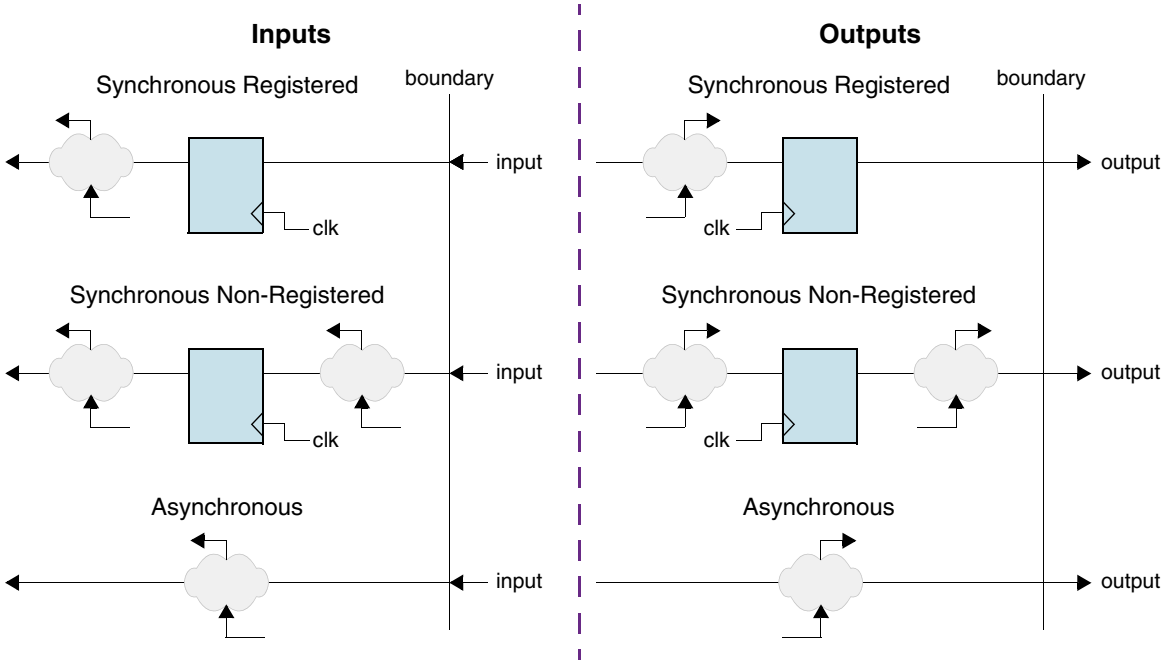
5.1 Naming and Description Conventions

Signals have the same names as in AMBA and D-PHY specifications for APB and PPI interfaces.

The signal name description describes the function of each signal and the type, that can be:

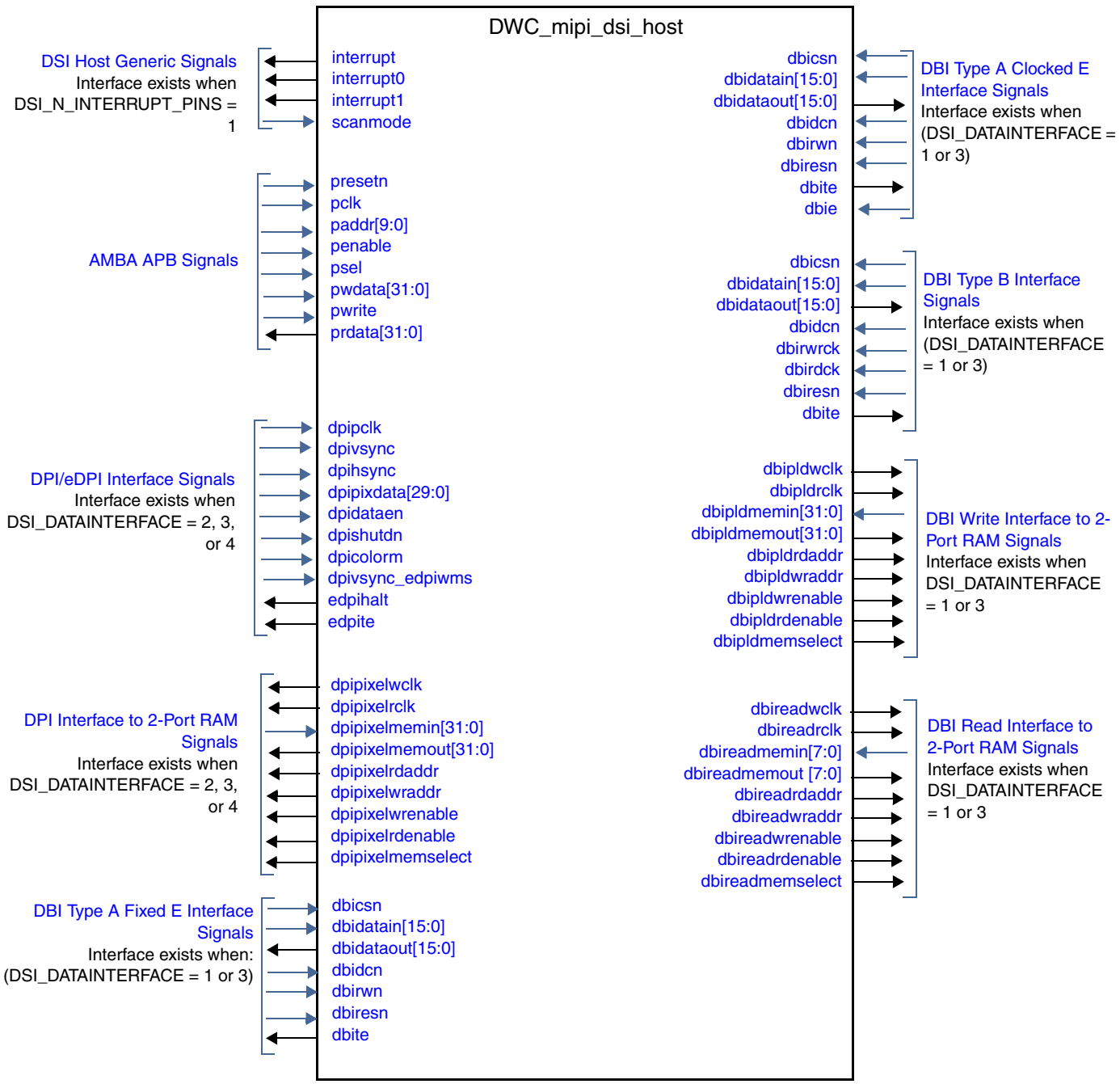
- **Synchronous:** The signal is asserted or de-asserted with respect to a clock edge.
- **Asynchronous:** The signal is not asserted or de-asserted with respect to a clock edge.
- **Registered:** The signal is captured (or launched) directly at the macro boundary with no intermediate logic between boundary and the capturing (or launching) flip-flop.

Figure 5-1 Synchronous and Asynchronous Signals



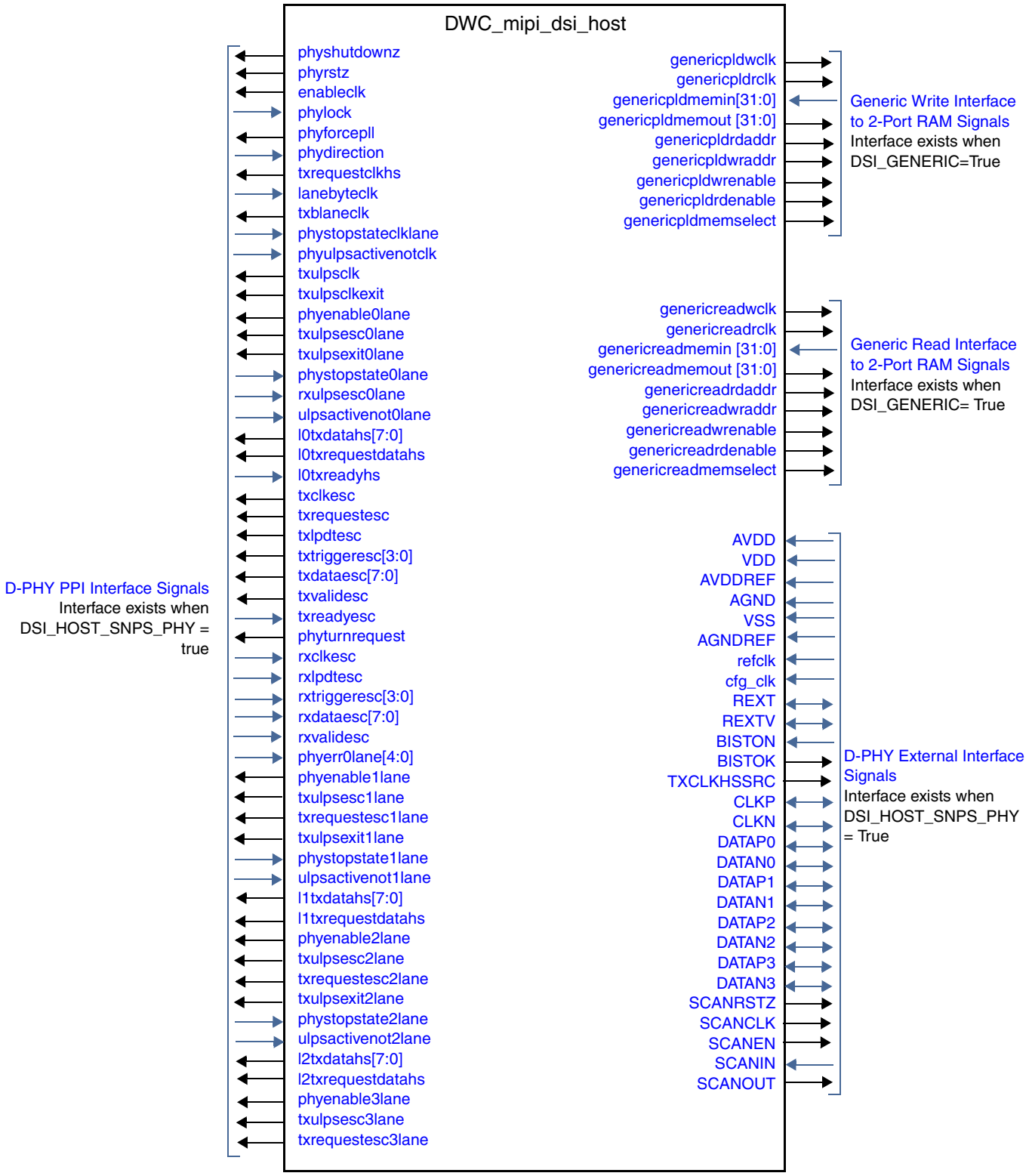
5.2 Top-Level I/O Diagram

Figure 5-2 Top-Level I/O Diagram



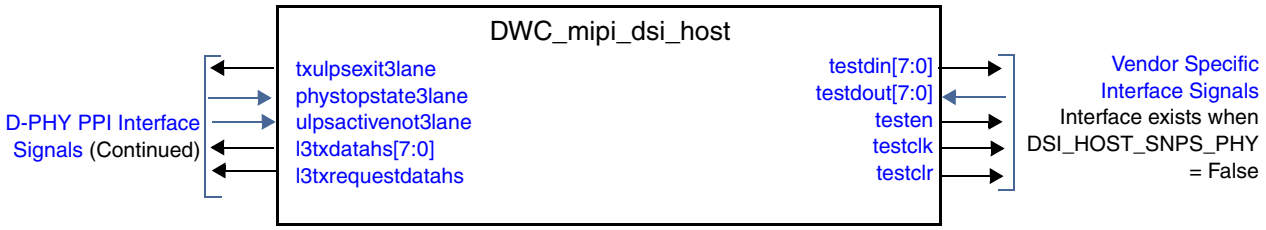
Continued on next page

Figure 5-2 Top-Level I/O Diagram



Continued on next page

Figure 5-2 Top-Level I/O Diagram



5.3 Signal Descriptions

The following sections provide a detailed description of the DWC_mipi_dsi_host signals.

5.3.1 DSI Host Generic Signals

The coreConsultant tool allows an interrupt pin configuration with the following values:

- 0: The DWC_mipi_dsi_host does not have an interrupt pin for notification of error although the system can check for error by reading error status registers, INT_ST0 and INT_ST1. These registers are cleared on read.
- 1: The DWC_mipi_dsi_host has an interrupt pin 'interrupt' that is set high when an error occurs either in the INT_ST0 register or in the INT_ST1 register. These registers are cleared on read, clearing the interrupt if the register that generated the error is read.
- 2: The DWC_mipi_dsi_host has two interrupts associated with each error status register. The interrupt0 and interrupt1 pins are respectively associated with the INT_ST0 and INT_ST1 registers. This mechanism allows faster association of an interrupt with the status register that generates the error. These registers are automatically cleared on read, clearing the associated interrupts.

Figure 5-3 DSI Host Generic Signals

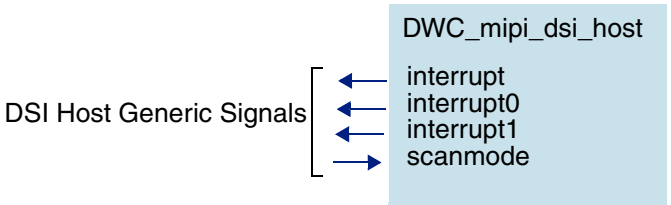


Table 5-1 Image Data Interface Signals

Name	Width	I/O	Description
interrupt	1 bit	O	Interruption Signal Active high for registers ERROR_ST0 and ERROR_ST1. **Dependency**: DSI_N_INTERRUPT_PINS = 1 **Active State**: High **Registered**: No. This is a direct OR of all the status registers and the MASK configuration. **Synchronous to**: pclk
interrupt0	1 bit	O	Interruption Signal Active high for register ERROR_ST0. **Dependency**: DSI_N_INTERRUPT_PINS = 2 **Active State**: High **Registered**: No. This is a direct OR of all the status registers and the MASK configuration. **Synchronous to**: pclk
interrupt1	1 bit	O	Interruption Signal Active high for register ERROR_ST1. **Dependency**: DSI_N_INTERRUPT_PINS = 2 **Active State**: High **Registered**: No. This is a direct OR of all the status registers and the MASK configuration. **Synchronous to**: pclk
scanmode	1 bit	I	Scan Mode Activation Signal **Dependency**: None **Active State**: High **Registered**: No **Synchronous to**: Asynchronous

5.3.2 AMBA APB Signals

Figure 5-4 AMBA Slave Interface Signals

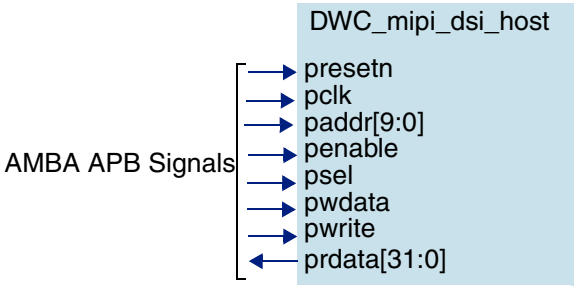


Table 5-2 AMBA Slave Interface Signals

Name	Width	I/O	Description
presetn	1 bit	I	APB Asynchronous Reset Signal This signal acts as global reset. **Dependency**: None **Active State**: Low **Registered**: No **Synchronous to**: The signal is asserted asynchronously, but is deasserted synchronously after the rising edge of pclk. The synchronization must be provided externally to this component.
pclk	1 bit	I	APB Clock Signal **Dependency**: None **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
paddr[9:0]	10 bits	I	APB Address Bus **Dependency**: None **Active State**: N/A **Registered**: No. The address is directly used to map the register used for reading or writing. **Synchronous to**: pclk
penable	1 bit	I	APB Enable Signal **Dependency**: None **Active State**: High **Registered**: No. This signal is directly used to enable read and write operations in the addressed register. **Synchronous to**: pclk

Table 5-2 AMBA Slave Interface Signals (Continued)

Name	Width	I/O	Description
psel	1 bit	I	APB Slave Selection Signal **Dependency**: None **Active State**: High **Registered**: No **Synchronous to**: pclk
pwdata[31:0]	32 bits	I	APB Write Data Bus **Dependency**: None **Active State**: N/A **Registered**: No **Synchronous to**: pclk
pwrite	1 bit	I	APB Write Enable Signal **Dependency**: None **Active State**: High **Registered**: No **Synchronous to**: pclk
prdata[31:0]	32 bits	O	APB Read Data Bus **Dependency**: None **Active State**: High **Registered**: Yes **Synchronous to**: pclk

5.3.3 DPI/eDPI Interface Signals

Figure 5-5 DPI/eDPI Interface Signals

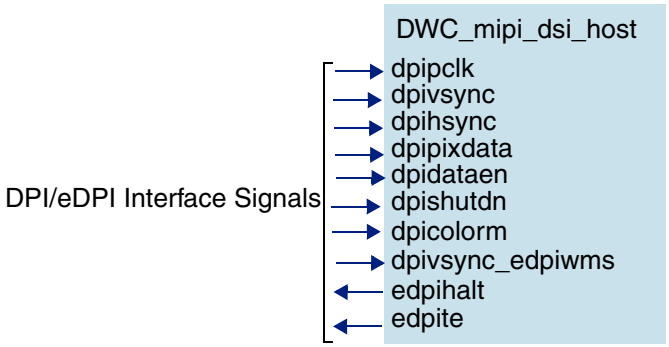


Table 5-3 DPI/eDPI Interface Signals

Name	Width	I/O	Description
dpipclk	1 bit	I	Input Pixel Clock Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
dpivsync	1 bit	I	Vertical Synchronism Signal **Dependency**: DSI_DATAINTERFACE = 2 or 3 **Active State**: Programmable **Registered**: No. Polarity change is applied at the signal before registering. **Synchronous to**: dpipclk
dpihsync	1 bit	I	Horizontal Synchronism Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: Programmable **Registered**: No. Polarity change is applied at the signal before registering. **Synchronous to**: dpipclk
dpipixdata[29:0]	30 bits	I	Video Data Video data is delivered one pixel per clock cycle. **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: No. Data is blocked with the dpidataen signal to avoid noise input. **Synchronous to**: dpipclk

Table 5-3 DPI/eDPI Interface Signals (Continued)

Name	Width	I/O	Description
dpidataen	1 bit	I	Video Data Enable Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: Programmable **Registered**: No. Polarity change is applied at the signal before registering. **Synchronous to**: dpipclk
dpishutdn	1 bit	I	Control Signal It is used to shutdown the display. **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: Programmable **Registered**: No. Polarity change is applied at the signal before registering. **Synchronous to**: dpipclk
dpicolorm	1 bit	I	Control Signal It is used to switching between normal color and reduced color mode. **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: Programmable **Registered**: No. Polarity change is applied at the signal before registering. **Synchronous to**: dpipclk
dpivsync_edpiwms	1 bit	I	Write Memory Start Signal **Dependency**: DSI_DATAINTERFACE = 4 **Active State**: High **Registered**: No **Synchronous to**: dpipclk
edpihalt	1 bit	O	Halt Indication on Video Interface **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: High **Registered**: No. The halts from the command and payload FIFOs are OR'ed in the Command mode and then multiplexed with the halt from the Video mode. **Synchronous to**: dpipclk
edpite	1 bit	O	Tearing Effect Indication This signal is asserted for one dpipclk cycle. **Dependency**: DSI_DATAINTERFACE = 4 **Active State**: High **Registered**: Yes **Synchronous to**: dpipclk

5.3.4 DPI Interface to 2-Port RAM Signals

The DPI interface is optional and can be configured in coreConsultant. If this interface is not selected, the DPI pins interface, RAM connections, and the associated logic are removed from the DWC_mipi_dsi_host.

If the DPI interface is selected, the address range for the 2-Port RAM needs to be configured based on system requirements.

Figure 5-6 DPI Interface to 2-Port RAM Signals

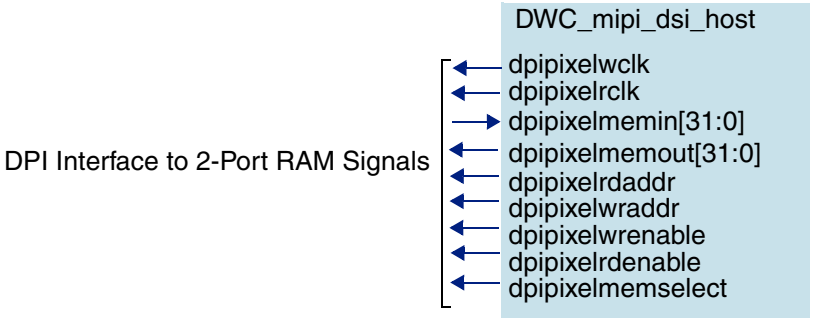


Table 5-4 DPI Interface to 2-Port RAM Signals

Name	Width	I/O	Description
dpipixelwclk	1 bit	O	Memory Write Clock Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: No **Synchronous to**: N/A
dpipixelrclk	1 bit	O	Memory Read Clock Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: No **Synchronous to**: N/A
dpipixelmemin[31:0]	32 bits	I	Memory Data Read Bus When read enable is active, this bus receives the memory data pointed by the address. **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: N/A **Synchronous to**: lanebyteclk
dpipixelmemout[31:0]	32 bits	O	Memory Data Write Bus **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: No **Synchronous to**: dpipclk

Table 5-4 DPI Interface to 2-Port RAM Signals (Continued)

Name	Width	I/O	Description
dpipixelrdaddr		O	Memory Read Address Bus The memory size is parameterizable. **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
dpipixelwraddr		O	Memory Write Address Bus The memory size is parameterizable. **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: N/A **Registered**: Yes **Synchronous to**: dpipclk
dpipixelwrenable	1 bit	O	Memory Write Enable Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: High **Registered**: No **Synchronous to**: dpipclk
dpipixelrdenable	1 bit	O	Memory Read Enable Signal **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
dpipixelmemselect	1 bit	O	Memory Interface Select **Dependency**: DSI_DATAINTERFACE = 2, 3, or 4 **Active State**: High **Registered**: No **Synchronous to**: dpipclk

5.3.5 DBI Type A Fixed E Interface Signals

Figure 5-7 DBI Type A Fixed E Interface Signals

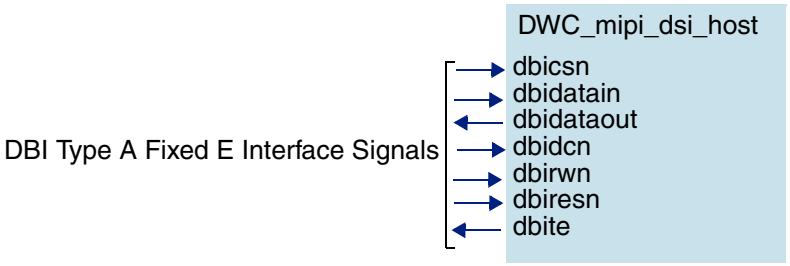


Table 5-5 DBI Type A Fixed E Interface Signals

Name	Width	I/O	Description
dbicsn	1 bit	I	DBI Chip Select The DWC_mipi_dsi_host writes the data at the rising edge or reads at the falling edge. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: Asynchronous
dbidatain[15:0]	16 bits	I	DBI Data Input Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbicsn
dbidataout[15:0]	16 bits	O	DBI Data Output Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbicsn
dbidcn	1 bit	I	DBI Data/Command When the signal is high, data is indicated and when low, command is indicated. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbicsn

Table 5-5 DBI Type A Fixed E Interface Signals (Continued)

Name	Width	I/O	Description
dbirwn	1 bit	I	DBI Read/Write The DWC_mipi_dsi_host writes the data at the rising edge or reads at the falling edge. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbicsn
dbiresn	1 bit	I	DBI Reset Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: Low **Registered**: No **Synchronous to**: Asynchronous
dbite	1 bit	O	DBI Tear Effect **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk

5.3.6 DBI Type A Clocked E Interface Signals

Figure 5-8 DBI Type A Clocked E Interface Signals

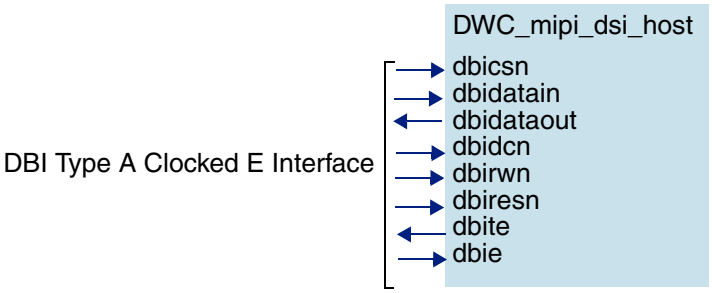


Table 5-6 DBI Type A Clocked E Interface Signals

Name	Width	I/O	Description
dbicsn	1 bit	I	DBI Chip Select **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: Low **Registered**: Yes **Synchronous to**: dbie
dbidatain[15:0]	16 bits	I	DBI Data Input Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbie
dbidataout[15:0]	16 bits	O	DBI Data Output Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbie
dbidcn	1 bit	I	DBI Data/Command When the signal is high, data is indicated and when low, command is indicated. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbie

Table 5-6 DBI Type A Clocked E Interface Signals (Continued)

Name	Width	I/O	Description
dbirwn	1 bit	I	DBI Read/Write When this signal is high, a write operation is performed and when it is low, a read operation is performed. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbie
dbiresn	1 bit	I	DBI Reset Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: Low **Registered**: No **Synchronous to**: Asynchronous
dbite	1 bit	O	DBI Tear Effect **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
dbie	1 bit	I	DBI E Clock The DWC_mipi_dsi_host reads the data at the falling edge or writes at the rising edge. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: Asynchronous

5.3.7 DBI Type B Interface Signals

Figure 5-9 DBI Type B Interface Signals

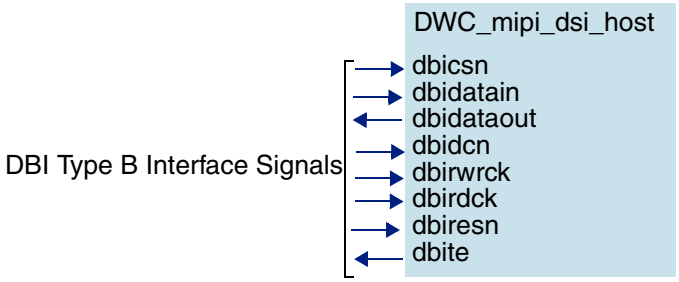


Table 5-7 DBI Type B Interface Signals

Name	Width	I/O	Description
dbicsn	1 bit	I	DBI Chip Select **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: Low **Registered**: Yes **Synchronous to**: dbirwrck
dbidatain[15:0]	16 bits	I	DBI Data Input Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbirwrck
dbidataout[15:0]	16 bits	O	DBI Data Output Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3) **Active State**: N/A **Registered**: Yes **Synchronous to**: dbirdck
dbidcn	1 bit	I	DBI Data/Command When the signal is high, data is indicated and when low, command is indicated. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: dbirwrck and dbirdck
dbirwrck	1 bit	I	DBI Write The DWC_mipi_dsi_host reads the information at the rising edge. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: Asynchronous

Table 5-7 DBI Type B Interface Signals (Continued)

Name	Width	I/O	Description
dbirdck	1 bit	I	DBI Read The DWC_mipi_dsi_host writes the information at the falling edge. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: Asynchronous
dbiresn	1 bit	I	DBI Reset Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: Low **Registered**: No **Synchronous to**: Asynchronous
dbite	1 bit	O	DBI Tear Effect **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk

5.3.8 DBI Write Interface to 2-Port RAM Signals

Figure 5-10 DBI Write Interface to 2-Port RAM Signals

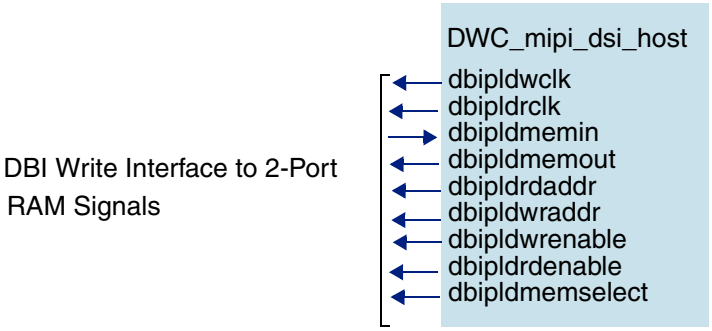


Table 5-8 DBI Write Interface to 2-Port RAM Signal Descriptions

Name	Width	I/O	Description
dbipldwclk	1 bit	O	DBI PLD WR Memory Write Clock Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: N/A
dbipldrclk	1 bit	O	DBI PLD WR Memory Read Clock Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: N/A
dbipldmemin[31:0]	32 bits	I	DBI PLD WR Memory Data Read Bus When read enable is active, this signal outputs the data pointed by the address. **Dependency**: DSI_DATAINTERFACE = 2 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: lanebyteclk
dbipldmemout[31:0]	32 bits	O	DBI PLD WR Memory Data Write Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: lanebyteclk

Table 5-8 DBI Write Interface to 2-Port RAM Signal Descriptions (Continued)

Name	Width	I/O	Description
dbipldrdaddr		O	DBI PLD WR Memory Read Address Bus Configurable RAM size. **Dependency**: DSI_DATAINTERFACE = 1 or 3. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_DBICMDADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
dbipldwraddr	1 bit	O	DBI PLD WR Memory Write Address Bus Configurable RAM size. **Dependency**: DSI_DATAINTERFACE = 1 or 3. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_DBICMDADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
dbipldwrenable	1 bit	O	DBI PLD WR Memory Write Enable Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
dbipldrdenable	1 bit	O	DBI PLD WR Memory Read Enable Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
dbipldmemselect	1 bit	O	DBI PLD WR Memory Select **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk

5.3.9 DBI Read Interface to 2-Port RAM Signals

If DBI interface is not selected, its interface pins, RAM connections, and the associated logic are removed from core.

If the DBI interface is selected, the address range for the read and the write 2-Port RAMs needs to be configured based on system requirements.

Figure 5-11 DBI Read Interface to 2-Port RAM Signals

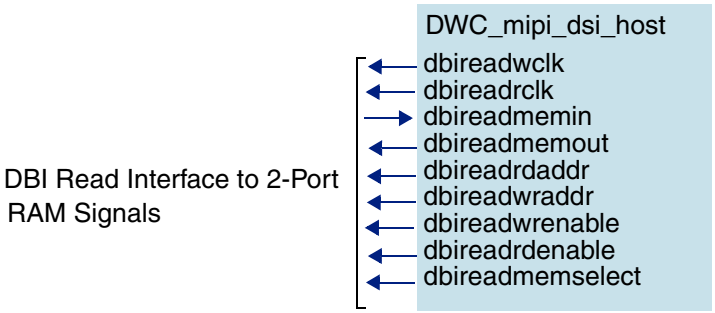


Table 5-9 DBI Read Interface to 2-Port RAM Signals

Name	Width	I/O	Description
dbireadwclk	1 bit	O	DBI PLD RD Memory Write Clock Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: N/A
dbireadrclk	1 bit	O	DBI PLD RD Memory Read Clock Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: N/A
dbireadmemin[7:0]	8 bits	I	DBI PLD RD Memory Data Read Bus When read enable is active, this signal outputs the data pointed by the address. **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
dbireadmemout [7:0]	8 bits	O	DBI PLD RD Memory Data Write Bus **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: N/A **Registered**: No **Synchronous to**: lanebyteclk

Table 5-9 DBI Read Interface to 2-Port RAM Signals (Continued)

Name	Width	I/O	Description
dbireadrdaddr	DBIREADPLDADDR WIDTH bits	O	DBI PLD RD Memory Read Address Bus Configurable RAM size. **Dependency**: DSI_DATAINTERFACE = 1 or 3. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_ DBIREADPLDADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
dbireadwraddr	DBIREADPLDADDR WIDTH bits	O	DBI PLD RD Memory Write Address Bus Configurable RAM size. **Dependency**: DSI_DATAINTERFACE = 1 or 3. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_ DBIREADPLDADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
dbireadwrenable	1 bit	O	DBI PLD RD Memory Write Enable Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
dbireadrdenable	1 bit	O	DBI PLD RD Memory Read Enable Signal **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
dbireadmemselect	1 bit	O	DBI PLD RD Memory Select **Dependency**: DSI_DATAINTERFACE = 1 or 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk

5.3.10 Generic Write Interface to 2-Port RAM Signals

The Generic interface is optional and can be configured in coreConsultant. If the Generic interface is not selected, its 2-Port RAM connection pins and associated logic are removed from the DWC_mipi_dsi_host. The Generic interface is accessible for write or read through the APB AMBA interface.

Figure 5-12 Generic Write Interface to 2-Port RAM Signals

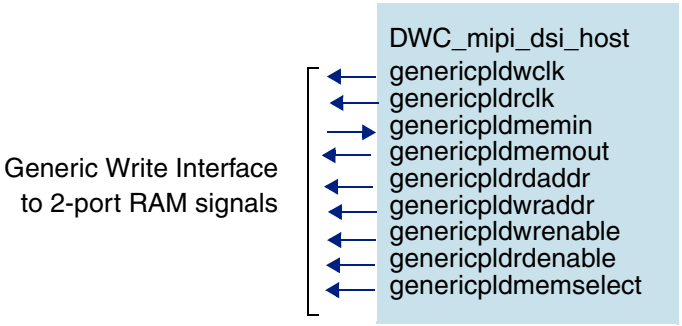


Table 5-10 Generic Write Interface to 2-Port RAM Signals

Name	Width	I/O	Description
genericpldwclk	1 bit	O	Generic PLD WR Memory Write Clock Signal **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: pclk
genericpldrclk	1 bit	O	Generic PLD WR Memory Read Clock Signal **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: lanebyteclk
genericpldmemin[31:0]	32 bits	I	Generic PLD WR Memory Data Read Bus When read enable is active, this signal outputs the data pointed by the address. **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: lanebyteclk
genericpldmemout [31:0]	32 bits	O	Generic PLD WR Memory Data Write Bus **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: pclk

Table 5-10 Generic Write Interface to 2-Port RAM Signals (Continued)

Name	Width	I/O	Description
genericpldrdaddr	DSI_GENERICPLDADDRDEPTH	O	Generic PLD WR Memory Read Address Bus Configurable RAM size. **Dependency**: DSI_GENERIC = True. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_GENERICPLDADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
genericpldwraddr	DSI_GENERICPLDADDRDEPTH	O	Generic PLD WR Memory Write Address Bus Configurable RAM size. **Dependency**: DSI_GENERIC = True. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_GENERICPLDADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
genericpldwrenable	1 bit	O	Generic PLD WR Memory Write Enable Signal **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: pclk
genericpldrdenable	1 bit	O	Generic PLD WR Memory Read Enable Signal **Dependency**: DSI_GENERIC = True **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
genericpldmemselect	1 bit	O	Generic PLD WR Memory Select Signal **Dependency**: DSI_GENERIC = True **Active State**: High **Registered**: No **Synchronous to**: pclk

5.3.11 Generic Read Interface to 2-Port RAM Signals

The Generic interface is optional and can be configured in coreConsultant. If the Generic interface is not selected, its 2-Port RAM connection pins and associated logic are removed from the DWC_mipi_dsi_host. The Generic interface is accessible for write or read through the APB AMBA interface.

Figure 5-13 Generic Read Interface to 2-Port RAM Signals

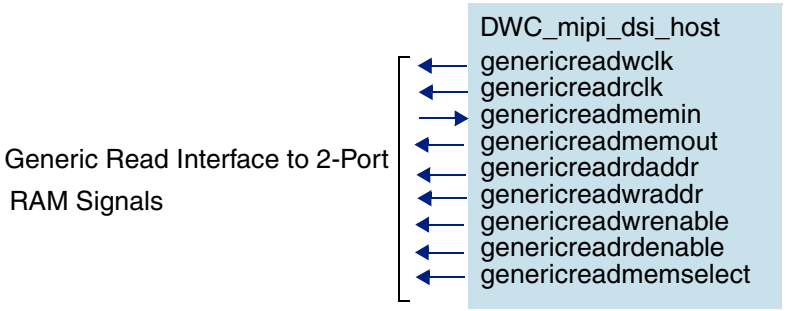


Table 5-11 Generic Read Interface to 2-Port RAM Signals

Name	Width	I/O	Description
genericreadwclk	1 bit	O	Generic PLD RD Memory Write Clock Signal **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: N/A
genericreadrclk	1bit	O	Generic PLD RD Memory Read Clock Signal **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: N/A
genericreadmemin [31:0]	32 bits	I	Generic PLD RD Memory Data Read Bus **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: pclk
genericreadmemout [31:0]	32 bits	O	Generic PLD RD Memory Data Write Bus **Dependency**: DSI_GENERIC = True **Active State**: N/A **Registered**: No **Synchronous to**: lanebyteclk

Table 5-11 Generic Read Interface to 2-Port RAM Signals (Continued)

Name	Width	I/O	Description
genericreadrdaddr	DSI_GENERICREADADDRWIDTH bits	O	Generic PLD RD Memory Read Address Bus Configurable RAM size. **Dependency**: DSI_GENERIC = True. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_GENERICREADADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
genericreadwraddr	DSI_GENERICREADADDRWIDTH bits	O	Generic PLD RD Memory Write Address Bus Configurable RAM size. **Dependency**: DSI_GENERIC = True. Bus width is determined in coreConsultant by the base 2 logarithm of the address depth (Parameter DSI_GENERICREADADDRDEPTH). **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
genericreadwrenable	1 bit	O	Generic PLD RD Memory Write Enable Signal **Dependency**: DSI_GENERIC = True **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
genericreadrdenable	1 bit	O	Generic PLD RD Memory Read Enable Signal **Dependency**: DSI_GENERIC = True **Active State**: High **Registered**: No **Synchronous to**: pclk
genericreadmemselect	1 bit	O	Generic PLD RD Memory Select **Dependency**: DSI_GENERIC = True **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk

5.3.12 D-PHY PPI Interface Signals

The D-PHY PPI Interface is present if the Synopsys D-PHY is not selected in coreConsultant. This section describes the signals that interface with a D-PHY through a standard PPI interface. When configuring the DWC_mipi_dsi_host, there is also an option to select the number of lanes in the DWC_mipi_dsi_host. One clock lane and one data lane (Lane0) with TX HS features are always required. Data lanes can be up to a maximum of four. Signals associated with the lanes that are not configured, are also removed.

Figure 5-14 D-PHY PPI Interface Signals

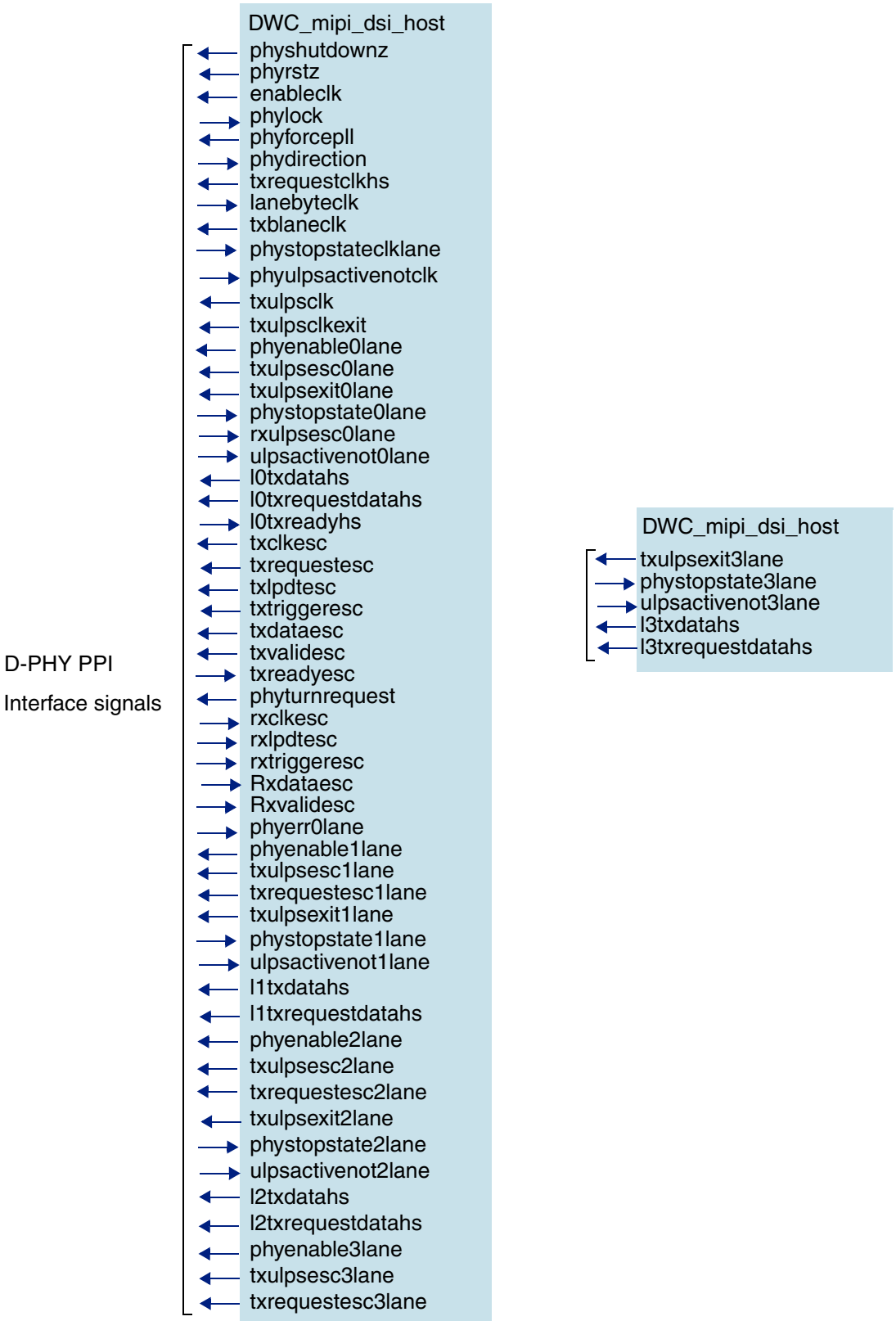


Table 5-12 D-PHY PPI Interface Signals

Name	Width	I/O	Description
D-PHY Configuration and Status Signals			
physhutdownz	1 bit	O	D-PHY Digital and Analog Shut Down **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: Low **Registered**: Yes **Synchronous to**: pclk
phyrstz	1 bit	O	D-PHY Reset **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: Low **Registered**: Yes **Synchronous to**: pclk
enableclk	1 bit	O	D-PHY Enable Clock Lane Generation **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: pclk
phylock	1 bit	I	D-PHY PLL Lock Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
phyforcepll	1 bit	O	D-PHY Force PLL On **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: pclk
phydirection	1 bit	I	D-PHY Current Direction of Lane 0 Interconnection ■ 0: TX ■ 1: RX **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
Clock Lane Control Signals			
txrequestclkhs	1 bit	O	D-PHY Request to Transmit High-Speed Clock **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
lanebyteclk	1 bit	I	D-PHY Lane Byte Clock (maximum of 125 MHz) **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
txblaneclk	1 bit	O	D-PHY Clock Lane Byte Return Signal Used for TX This is a duplicate of lanebyteclk. **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
phystopstateclklane	1 bit	I	D-PHY Clock Lane Stop State Notification **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
phyulpsactivenotclk	1 bit	I	This signal indicates that the clock lane is in the ULPM. **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: Low **Registered**: N/A **Synchronous to**: N/A
txulpsclk	1 bit	O	D-PHY Clock Lane Transmit ULPM **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
txulpsclkexit	1 bit	O	D-PHY Clock Lane Transmit Exit from ULPM **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
Lane 0 Signals			
phyenable0lane	1 bit	O	D-PHY Lane 0 Enable Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: No. It is a always constant, that is, 1. **Synchronous to**: Asynchronous
txulpsesc0lane	1 bit	O	D-PHY Data Lane 0 Transmit/Enter ULPM **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: No. **Synchronous to**: lanebyteclk
txulpsexit0lane	1bit	O	D-PHY Data Lane 0 Exit ULPM **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
phystopstate0lane	1 bit	I	D-PHY Lane 0 is in Stop State **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
rxulpsesc0lane	1 bit	I	Lane 0 ULPM Escape Code Receive **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
ulpsactivenot0lane	1 bit	I	Lane 0 ULPM Active **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: Low **Registered**: Yes **Synchronous to**: Asynchronous
l0txdatahs[7:0]	8 bits	O	Lane 0 TX High-Speed Transmit Data Output **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: Yes **Registered**: Yes **Synchronous to**: lanebyteclk

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
l0txrequestdatahs	1 bit	O	Lane 0 TX High-Speed Transmit Data Sending Request **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
l0txreadyhs	1 bit	I	Lane 0 TX High-Speed Transmit Ready **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
txclkesc	1 bit	O	Lane 0 LP TX Escape Mode Clock Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
txrequestesc	1 bit	O	Lane 0 LP TX Escape Mode Transmit Request **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txlpdtesc	1 bit	O	Lane 0 LP TX Escape Mode Transmit LP Data Request **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txtriggeresc[3:0]	4 bits	O	Lane 0 LP TX Escape Mode Transmission Triggers **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
txdataesc[7:0]	8 bits	O	Lane 0 LP TX Escape Mode LP Data Transmission **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: lanebyteclk
txvalidesc	1 bit	O	Lane 0 LP TX Escape Mode LP Data Valid Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Registered**: Yes **Synchronous to**: lanebyteclk

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
txreadyesc	1 bit	I	Lane 0 LP TX Escape Mode LP Data Transmit Ready Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
phyturnrequest	1 bit	O	D-PHY Turn Request Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
rxclkesc	1 bit	I	Lane 0 LP RX Escape Mode Clock Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: No **Synchronous to**: Asynchronous
rxlpdtesc	1 bit	I	Lane 0 LP RX Escape Mode LP Data Receive Mode **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: rxclkesc
rxtriggeresc[3:0]	4 bits	I	Lane 0 LP RX Escape Mode Received Triggers **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: rxclkesc
rxdataesc[7:0]	8 bits	I	Lane 0 LP RX Escape Mode Received Data **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: rxclkesc
rxvalidesc	1 bit	I	Lane 0 LP RX Escape Mode Valid Receive Data **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: rxclkesc

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
phyerr0lane[4:0]	5 bits	I	D-PHY Lane 0 Error Input Signals **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
Lane 1 Signals			
phyenable1lane	1 bit	O	D-PHY Lane 1 Enable Signal See D-PHY datasheet. **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: No **Synchronous to**: pclk
txulpsesc1lane	1 bit	O	D-PHY Data Lane 1 Transmit/Enter ULPM **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txrequestesc1lane	1 bit	O	D-PHY Data Lane 1 Request Transmit Escape Mode **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txulpsexit1lane	1 bit	O	D-PHY Data Lane 1 Exit ULPM **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
phystopstate1lane	1 bit	I	D-PHY Lane 1 is in Stop State **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
ulpsactivenot1lane	1 bit	I	Lane 1 ULPM Active **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: Low **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
l1txdatahs[7:0]	8 bits	O	Lane 1 TX High-Speed Transmit Data Output **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
l1txrequestdatahs	1 bit	O	Lane 1 TX High-Speed Transmit Data Sending Request **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
Lane 2 Signals			
phyenable2lane	1 bit	O	D-PHY Lane 2 Enable Signal See D-PHY datasheet. **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: **Registered**: No **Synchronous to**: pclk
txulpsesc2lane	1 bit	O	D-PHY Data Lane 2 Transmit/Enter ULPM **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txrequestesc2lane	1 bit	O	D-PHY Data Lane 2 Request Transmit Escape Mode **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
txulpsexit2lane	1 bit	O	D-PHY Data Lane 2 Exit ULPM **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
phystopstate2lane	1 bit	I	D-PHY Lane 2 is in Stop State **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
ulpsactivenot2lane	1 bit	I	Lane 2 ULPM Active **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: Low **Registered**: Yes **Synchronous to**: Asynchronous
l2txdatahs[7:0]	8 bits	O	Lane 2 TX High-Speed Transmit Data Output **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: **Registered**: Yes **Synchronous to**: lanebyteclk
l2txrequestdatahs	1 bit	O	Lane 2 TX High-Speed Transmit Data Sending Request **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES =3 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
Lane 3 Signals			
phyenable3lane	1 bit	O	D-PHY Lane 3 Enable Signal See D-PHY datasheet. **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: High **Registered**: No **Synchronous to**: pclk

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
txulpsesc3lane	1 bit	O	D-PHY Data Lane 3 Transmit/Enter ULPM **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txrequestesc3lane	1 bit	O	D-PHY Data Lane 3 Request Transmit Escape Mode **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
txulpsexit3lane	1 bit	O	D-PHY Data Lane 3 Exit ULPM **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: High **Registered**: No **Synchronous to**: lanebyteclk
phystopstate3lane	1 bit	I	D-PHY Lane 3 is in Stop State **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: High **Registered**: Yes **Synchronous to**: Asynchronous
ulpsactivenot3lane	1 bit	I	Lane 3 ULPM Active **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: Low **Registered**: Yes **Synchronous to**: Asynchronous
l3txdatahs[7:0]	8 bits	O	Lane 3 TX High-Speed Transmit Data Output **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: **Registered**: Yes **Synchronous to**: lanebyteclk

Table 5-12 D-PHY PPI Interface Signals (Continued)

Name	Width	I/O	Description
l3txrequestdatahs	1 bit	O	Lane 3 TX High-Speed Transmit Data Sending Request **Dependency**: DSI_HOST_SNPS_PHY = False and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: High **Registered**: Yes **Synchronous to**: lanebyteclk
PHY Parallel Test Control Signals			
testdin[7:0]	8 bits	O	D-PHY Test Data Output Port **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
testdout[7:0]	8 bits	I	D-PHY Test Data Input Port **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
testen	1 bit	O	D-PHY Test Enable **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: pclk
testclk	1 bit	O	D-PHY Test Clock Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
testclr	1 bit	O	D-PHY Test Clear Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: pclk

5.3.13 Vendor Specific Interface Signals

Figure 5-15 Vendor Specific Interface Signals

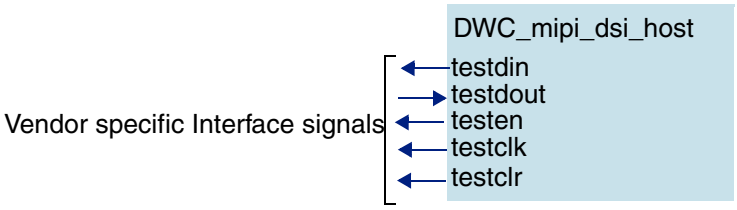


Table 5-13 Vendor Specific Interface Signals

Name	Width	I/O	Description
testdin[7:0]	8 bits	O	D-PHY Test Data Output Port **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
testdout[7:0]	8 bits	I	D-PHY Test Data Input Port **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
testen	1 bit	O	D-PHY Test Enable **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: pclk
testclk	1 bit	O	D-PHY Test Clock Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: N/A **Registered**: Yes **Synchronous to**: pclk
testclr	1 bit	O	D-PHY Test Clear Signal **Dependency**: DSI_HOST_SNPS_PHY = False **Active State**: High **Registered**: Yes **Synchronous to**: pclk

5.3.14 D-PHY External Interface Signals

The D-PHY External Interface is present if Synopsys D-PHY is selected. Table 5-14 provides the D-PHY pin list with the pins expected when an internal Synopsys D-PHY is selected in coreConsultant. The PPI interface described in "D-PHY PPI Interface Signals" is handled inside the DWC_mipi_dsi_host, and the connections between the D-PHY and the DSI protocol layer no longer exist at the top-level pinout.

Only the lane differential signals configured in coreConsultant appear as pins as in the case of PPI interface.

Figure 5-16 D-PHY External Interface Signals

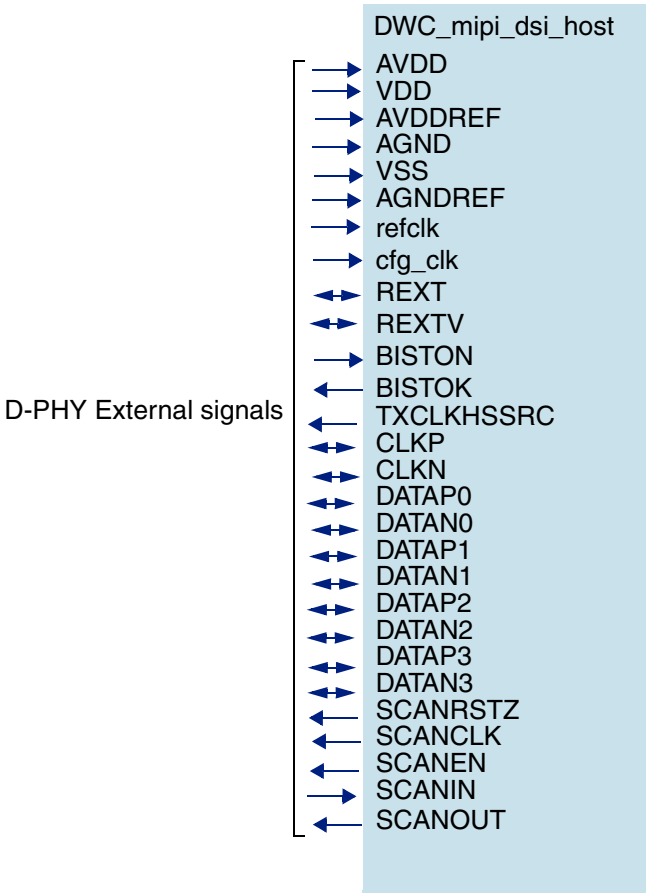


Table 5-14 D-PHY External Signals

Name	Width	I/O	Description
D-PHY Power Supply Signals			
AVDD	1 bit	I	D-PHY Analog Power Supply **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous

Table 5-14 D-PHY External Signals (Continued)

Name	Width	I/O	Description
VDD	1 bit	I	D-PHY Digital Power Supply **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
AVDDREF	1 bit	I	D-PHY Analog Supply for Reference Generator **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
AGND	1 bit	I	D-PHY Analog Supply Ground Return **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
VSS	1 bit	I	D-PHY Digital Supply Ground Return **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
AGNDREF	1 bit	I	D-PHY Analog Supply Ground Return for Reference Generator **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
D-PHY External Signals			
refclk	1 bit	I	D-PHY Reference Clock It is used for master-side serial clock generation in clock multiplying unit (PLL). **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous

Table 5-14 D-PHY External Signals (Continued)

Name	Width	I/O	Description
cfg_clk	1 bit	I	D-PHY Configuration Clock It is used for the initialization of the PHY. It is also used for exiting the ULPS state. **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
REXT	1 bit	I/O	D-PHY External Resistor Connection REXT and REXTV should be shorted at the chip pad connection. **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
REXTV	1 bit	I/O	D-PHY External Resistor Connection REXT and REXTV should be shorted at the chip pad connection. **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
BISTON	1 bit	I	PHY BIST On Indication **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: High **Registered**: N/A **Synchronous to**: SCANCLK
BISTOK	1 bit	O	PHY BIST OK Indication **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: High **Registered**: Yes **Synchronous to**: SCANCLK
TXCLKHSSRC	1 bit	O	D-PHY High-Speed Transmit Bit Clock with Frequency Identical to the Line Data Rate **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous

Table 5-14 D-PHY External Signals (Continued)

Name	Width	I/O	Description
CLKP	1 bit	I/O	Positive D-PHY Differential Clock Line Transceiver Output **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
CLKN	1 bit	I/O	Negative D-PHY Differential Clock Line Transceiver Output **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
DATAP0	1 bit	I/O	Positive D-PHY Differential Data Line Transceiver Output, Lane 0 **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
DATAN0	1 bit	I/O	Negative D-PHY Differential Data Line Transceiver Output, Lane 0 **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
DATAP1	1 bit	I/O	Positive D-PHY Differential Data Line Transceiver Output, Lane 1 **Dependency**: DSI_HOST_SNPS_PHY = True and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
DATAN1	1 bit	I/O	Negative D-PHY Differential Data Line Transceiver Output, Lane 1 **Dependency**: DSI_HOST_SNPS_PHY = True and DSI_HOST_NUMBER_OF_LANES = 2 **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
DATAP2	1 bit	I/O	Positive D-PHY Differential Data Line Transceiver Output, Lane 2 **Dependency**: DSI_HOST_SNPS_PHY = True and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous

Table 5-14 D-PHY External Signals (Continued)

Name	Width	I/O	Description
DATAN2	1 bit	I/O	Negative D-PHY Differential Data Line Transceiver Output, Lane 2 **Dependency**: DSI_HOST_SNPS_PHY = True and DSI_HOST_NUMBER_OF_LANES = 3 **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
DATAP3	1 bit	I/O	Positive D-PHY Differential Data Line Transceiver Output, Lane 3 **Dependency**: DSI_HOST_SNPS_PHY = True and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
DATAN3	1 bit	I/O	Negative D-PHY Differential Data Line Transceiver Output, Lane 3 **Dependency**: DSI_HOST_SNPS_PHY = True and DSI_HOST_NUMBER_OF_LANES = 4 **Active State**: N/A **Registered**: N/A **Synchronous to**: Asynchronous
D-PHY Scan Signals			
SCANRSTZ	1 bit	I	D-PHY Reset Signal for Scan Mode **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: Low **Registered**: No **Synchronous to**: SCANCLK
SCANCLK	1 bit	I	D-PHY Scan Clock Source for Scan Mode **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: N/A
SCANEN	1 bit	I	D-PHY Scan Enable ■ 1: Shift mode ■ 0: Capture mode **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: No **Synchronous to**: SCANCLK

Table 5-14 D-PHY External Signals (Continued)

Name	Width	I/O	Description
SCANIN	N_SCAN_CHAINS	I	D-PHY Scan In Serial Data Stream Input The macro should be in Scan mode. **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: Yes **Synchronous to**: SCANCLK
SCANOUT	N_SCAN_CHAINS	O	D-PHY Scan Out Serial Data Stream Output The macro should be in Scan mode. **Dependency**: DSI_HOST_SNPS_PHY = True **Active State**: N/A **Registered**: N/A **Synchronous to**: SCANCLK

Note The Scan interface for the D-PHY is dependent on the number of lanes presented in the Synopsys PHY. For details about the number of scan chains for a particular number of lanes, see *DesignWare Cores MIPI Bidir 4L D-PHY Databook*.

6 Register Descriptions

This chapter describes the DWC_mipi_dsi_host registers. It contains the following sections:

6.1 Register Memory Map

The DWC_mipi_dsi_host registers are addressable on 32-bit boundaries. Each unused bit or address location is reserved for future use and read back as 0.

Table 6-1 summarizes the register memory map for the DWC_mipi_dsi_host. The access type R stands for Read Only bit and R/W stands for Read/Write bit.

Table 6-1 DWC_mipi_dsi_host Memory Map

Register Name	Address Offset	Access Type	Description	Value after Reset
VERSION	0x000	R	Version of the DSI host controller	0x3132302A
PWR_UP	0x004	R/W	Core power-up	0x00000000
CLKMGR_CFG	0x008	R/W	Configuration of the internal clock dividers	0x00000000
DPI_VCID	0x00C	R/W	DPI virtual channel id	0x00000000
DPI_COLOR_CODING	0x010	R/W	DPI color coding	0x00000000
DPI_CFG_POL	0x014	R/W	DPI polarity configuration	0x00000000
DPI_LP_CMD_TIM	0x018	R/W	Low-power command timing configuration	0x00000000
DBI_VCID	0x01C	R/W	DBI virtual channel id	0x00000000
DBI_CFG	0x020	R/W	DBI interface configuration	0x00000000
DBI_PARTITIONING_EN	0x024	R/W	DBI partitioning enable	0x00000000
DBI_CMDSIZE	0x028	R/W	DBI command size configuration	0x00000000
PCKHDL_CFG	0x02C	R/W	Packet handler configuration	0x00000000
GEN_VCID	0x030	R/W	Generic interface virtual channel id	0x00000000
MODE_CFG	0x034	R/W	Video or Command mode selection	0x00000001
VID_MODE_CFG	0x038	R/W	Video mode configuration	0x00000000
VID_PKT_SIZE	0x03C	R/W	Video packet size	0x00000000
VID_NUM_CHUNKS	0x040	R/W	Number of chunks	0x00000000
VID_NULL_SIZE	0x044	R/W	Null packet size	0x00000000
VID_HSA_TIME	0x048	R/W	Horizontal Sync Active time	0x00000000
VID_HBP_TIME	0x04C	R/W	Horizontal Back Porch time	0x00000000
VID_HLINE_TIME	0x050	R/W	Line time	0x00000000
VID_VSA_LINES	0x054	R/W	Vertical Sync Active period	0x00000000

Table 6-1 DWC_mipi_dsi_host Memory Map (Continued)

Register Name	Address Offset	Access Type	Description	Value after Reset
VID_VBP_LINES	0x058	R/W	Vertical Back Porch period	0x00000000
VID_VFP_LINES	0x05C	R/W	Vertical Front Porch period	0x00000000
VID_VACTIVE_LINES	0x060	R/W	Vertical resolution	0x00000000
EDPI_CMD_SIZE	0x064	R/W	Size for eDPI packets	0x00000000
CMD_MODE_CFG	0x068	R/W	Command mode configuration	0x00000000
GEN_HDR	0x06C	R/W	Generic packet header configuration	0x00000000
GEN_PLD_DATA	0x070	R/W	Generic payload data in and out	0x00000000
CMD_PKT_STATUS	0x074	R	Command packet status	0x00001515
TO_CNT_CFG	0x078	R/W	Timeout timers configuration	0x00000000
HS_RD_TO_CNT	0x07C	R/W	Peripheral response timeout definition after high-speed read operations	0x00000000
LP_RD_TO_CNT	0x080	R/W	Peripheral response timeout definition after low-power read operations	0x00000000
HS_WR_TO_CNT	0x084	R/W	Peripheral response timeout definition after high-speed write operations	0x00000000
LP_WR_TO_CNT	0x088	R/W	Peripheral response timeout definition after low-power write operations	0x00000000
BTA_TO_CNT	0x08C	R/W	Peripheral response timeout definition after bus turnaround	0x00000000
SDF_3D	0x090	R/W	3D control configuration	0x00000000
LPCLK_CTRL	0x094	R/W	Low-power in clock lane	0x00000000
PHY_TMR_LPCLK_CFG	0x098	R/W	D-PHY timing configuration for the clock lane	0x00000000
PHY_TMR_CFG	0x09C	R/W	D-PHY timing configuration for data lanes	0x00000000
PHY_RSTZ	0x0A0	R/W	D-PHY reset control	0x00000000
PHY_IF_CFG	0x0A4	R/W	D-PHY interface configuration	DSI_HOST_NUMBER_OF_LANES-1
PHY_ULPS_CTRL	0x0A8	R/W	D-PHY Ultra Low-Power control	0x00000000
PHY_TX_TRIGGERS	0x0AC	R/W	D-PHY transmit triggers	0x00000000
PHY_STATUS	0x0B0	R	D-PHY PPI status interface	0x00000000

Table 6-1 DWC_mipi_dsi_host Memory Map (Continued)

Register Name	Address Offset	Access Type	Description	Value after Reset
PHY_TST_CTRL0	0x0B4	R/W	D-PHY test interface control 0	0x00000001
PHY_TST_CTRL1	0x0B8	R/W	D-PHY test interface control 1	0x00000000
INT_ST0	0x0BC	R	Interrupt status register 0	0x00000000
INT_ST1	0x0C0	R	Interrupt status register 1	0x00000000
INT_MSK0	0x0C4	R/W	Masks the interrupt generation triggered by the INT_ST0 register	0x001FFFFF
INT_MSK1	0x0C8	R/W	Masks the interrupt generation triggered by the INT_ST1 register	0x0003FFFF

6.2 Register and Field Descriptions

The section describes the data fields of the DWC_mipi_dsi_host registers.

6.2.1 VERSION

- **Description:** This register contains the version of the DWC_mipi_dsi_host.
- **Address Offset:** 0x000
- **Value after Reset:** 0x3132302A
- **Access:** Read

Table 6-2 Controller Version Identification Register

Bits	Name	Access Type	Description
31:0	VERSION	R	This field indicates the version of the DWC_mipi_dsi_host.

6.2.2 PWR_UP

- **Description:** This register controls the power-up of the DWC_mipi_dsi_host.
- **Address Offset:** 0x004
- **Value after Reset:** 0x0
- **Access:** Read/Write

Table 6-3 Power-Up Register

Bits	Name	Access Type	Description
31:1			Reserved
0	shutdownz	R/W	This bit configures the core either to power up or to reset. Shutdownz is the soft reset register. Its default value is 0. After the core configuration, to enable the DWC_mipi_dsi_host, set this register to 1. ■ 0: Reset ■ 1: Power-up

6.2.3 CLKMGR_CFG

- **Description:** This register configures the factor for the internal dividers to divide lanebyteclk for timeout purpose.
- **Address offset:** 0x008
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-4 Internal Clock Dividers Configuration Register

Bits	Name	Access Type	Description
31:16			Reserved
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 HS to LP and LP to HS transition error.
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.

6.2.4 DPI_VCID

- **Description:** This register configures the virtual channel id for DPI traffic.
- **Address offset:** 0x00C
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-5 DPI Virtual Channel ID Register

Bits	Name	Access Type	Description
31:2			Reserved
1:0	dpi_vcid	R/W	This field configures the DPI virtual channel id that is indexed to the Video mode packets.

6.2.5 DPI_COLOR_CODING

- **Description:** This register configures the DPI color coding.
- **Address offset:** 0x010
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-6 DPI Color Coding Register

Bits	Name	Access Type	Description
31:9			Reserved
8	loosely18_en	R/W	When set to 1, this bit activates loosely packed variant to 18-bit configurations.
7:4			Reserved
3:0	dpi_color_coding	R/W	This field configures the DPI color coding as follows: ■ 0000: 16-bit configuration 1 ■ 0001: 16-bit configuration 2 ■ 0010: 16-bit configuration 3 ■ 0011: 18-bit configuration 1 ■ 0100: 18-bit configuration 2 ■ 0101: 24-bit ■ 0110: 20-bit YCbCr 4:2:2 loosely packed ■ 0111: 24-bit YCbCr 4:2:2 ■ 1000: 16-bit YCbCr 4:2:2 ■ 1001: 30-bit ■ 1010: 36-bit ■ 1011-1111: 12-bit YCbCr 4:2:0

6.2.6 DPI_CFG_POL

- **Description:** This register configures the polarity of the DPI signals.
- **Address offset:** 0x014
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-7 DPI Polarity Configuration Register

Bits	Name	Access Type	Description
31:5			Reserved
4	colorm_active_low	R/W	When set to 1, this bit configures the color mode pin (dpicolorm) as active low.
3	shutd_active_low	R/W	When set to 1, this bit configures the shutdown pin (dpishutdn) as active low.
2	hsync_active_low	R/W	When set to 1, this bit configures the horizontal synchronism pin (dpihsync) as active low.
1	vsync_active_low	R/W	When set to 1, this bit configures the vertical synchronism pin (dpivsync) as active low.
0	dataen_active_low	R/W	When set to 1, this bit configures the data enable pin (dpidataen) as active low.

6.2.7 DPI_LP_CMD_TIM

- **Description:** This register configures the timing for the low-power commands sent while in video mode.
- **Address offset:** 0x018
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-8 Low-Power Command Timing Configuration Register

Bits	Name	Access Type	Description
31:24			Reserved

Table 6-8 Low-Power Command Timing Configuration Register (Continued)

Bits	Name	Access Type	Description
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.
15:8			Reserved
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.

6.2.8 DBI_VCID

- **Description:** This register configures virtual channel id for DBI traffic.
- **Address offset:** 0x01C
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-9 DBI Virtual Channel Id Register

Bits	Name	Access Type	Description
31:2			Reserved
1:0	dbi_vcid	R/W	This field configures the virtual channel id that is indexed to the DCS packets from DBI.

6.2.9 DBI_CFG

- **Description:** This register configures the bit width of pixels for DBI.
- **Address offset:** 0x020
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-10 DBI Interface Configuration Register

Bits	Name	Access Type	Description
31:18			Reserved

Table 6-10 DBI Interface Configuration Register (Continued)

Bits	Name	Access Type	Description
17:16	lut_size_conf	R/W	This field configures the size used to transport the write Lut commands as follows: ■ 00: 16-bit color display ■ 01: 18-bit color display ■ 10: 24-bit color display ■ 11: 16-bit color display
15:12			Reserved
11:8	out_dbi_conf	R/W	This field configures the DBI output pixel data as follows: ■ 0000: 8-bit 8 bpp ■ 0001: 8-bit 12 bpp ■ 0010: 8-bit 16 bpp ■ 0011: 8-bit 18 bpp ■ 0100: 8-bit 24 bpp ■ 0101: 9-bit 18 bpp ■ 0110: 16-bit 8 bpp ■ 0111: 16-bit 12 bpp ■ 1000: 16-bit 16 bpp ■ 1001: 16-bit 18 bpp, option 1 ■ 1010: 16-bit 18 bpp, option 2 ■ 1011: 16-bit 24 bpp, option 1 ■ 1100: 16-bit 24 bpp, option 2
7:4			Reserved

Table 6-10 DBI Interface Configuration Register (Continued)

Bits	Name	Access Type	Description
3:0	in_dbi_conf	R/W	This field configures the DBI input pixel data as follows: ■ 0000: 8-bit 8 bpp ■ 0001: 8-bit 12 bpp ■ 0010: 8-bit 16 bpp ■ 0011: 8-bit 18 bpp ■ 0100: 8-bit 24 bpp ■ 0101: 9-bit 18 bpp ■ 0110: 16-bit 8 bpp ■ 0111: 16-bit 12 bpp ■ 1000: 16-bit 16 bpp ■ 1001: 16-bit 18 bpp, option 1 ■ 1010: 16-bit 18 bpp, option 2 ■ 1011: 16-bit 24 bpp, option 1 ■ 1100: 16-bit 24 bpp, option 2

6.2.10 DBI_PARTITIONING_EN

- **Description:** This register configures whether the DWC_mipi_dsi_host is to partition the DBI traffic automatically.
- **Address offset:** 0x024
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-11 DBI Partitioning Enable Register

Bits	Name	Access Type	Description
31:1			Reserved
0	partitioning_en	R/W	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.

6.2.11 DBI_CMDSIZE

- **Description:** This register configures the command size and the size for automatic partitioning of the DBI packets.
- **Address offset:** 0x028
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-12 DBI Command Size Configuration Register

Bits	Name	Access Type	Description
31:16	allowed_cmd_size	R/W	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.
15:0	wr_cmd_size	R/W	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.

6.2.12 PCKHDL_CFG

- **Description:** This register configures how EoTp, BTA, CRC, and ECC are to be used, to meet the peripheral's characteristics.
- **Address offset:** 0x02C
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-13 Packet Handler Configuration Register

Bits	Name	Access Type	Description
31:5			Reserved
4	crc_rx_en	R/W	When set to 1, this bit enables the CRC reception and error reporting. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1. Otherwise, this bit is reserved.
3	ecc_rx_en	R/W	When set to 1, this bit enables the ECC reception, error correction, and reporting.
2	bta_en	R/W	When set to 1, this bit enables the Bus Turn-Around (BTA) request.

Table 6-13 Packet Handler Configuration Register (Continued)

Bits	Name	Access Type	Description
1	eotp_rx_en	R/W	When set to 1, this bit enables the EoTp reception.
0	eotp_tx_en	R/W	When set to 1, this bit enables the EoTp transmission.

6.2.13 GEN_VCID

- **Description:** This register configures the virtual channel id of the read responses to store and return to the Generic interface.
- **Address offset:** 0x030
- **Dependency:** DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-14 Generic Interface Virtual Channel Id Register

Bits	Name	Access Type	Description
31:2			Reserved
1:0	gen_vcid_rx	R/W	This field indicates the Generic interface read-back virtual channel identification.

6.2.14 MODE_CFG

- **Description:** This register configures the mode of operation - Video mode or Command mode (Commands can be sent even in video mode too).
- **Address offset:** 0x034
- **Value after reset:** 0x00000001
- **Access:** Read/Write

Table 6-15 Video or Command Mode Selection Register

Bits	Name	Access Type	Description
31:1			Reserved
0	cmd_video_mode	R/W	This bit configures the operation mode: ■ 0: Video mode ■ 1: Command mode

6.2.15 VID_MODE_CFG

- **Description:** This register configures several aspects of the Video mode operation such as the transmission mode, switching to low-power in the middle of a frame, enabling acknowledge, and whether to send the commands in low-power.
- **Address offset:** 0x038
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-16 Video Mode Configuration Register

Bits	Name	Access Type	Description
31:16			Reserved
15	lp_cmd_en	R/W	When set to 1, this bit enables the command transmission only in low-power mode.
14	frame_bta_ack_en	R/W	When set to 1, this bit enables the request for an acknowledge response at the end of a frame.
13	lp_hfp_en	R/W	When set to 1, this bit enables the return to low-power inside the HFP period when timing allows.
12	lp_hbp_en	R/W	When set to 1, this bit enables the return to low-power inside the HBP period when timing allows.
11	lp_vact_en	R/W	When set to 1, this bit enables the return to low-power inside the VACT period when timing allows.
10	lp_vfp_en	R/W	When set to 1, this bit enables the return to low-power inside the VFP period when timing allows.
9	lp_vbp_en	R/W	When set to 1, this bit enables the return to low-power inside the VBP period when timing allows.
8	lp_vsa_en	R/W	When set to 1, this bit enables the return to low-power inside the VSA period when timing allows.
7:2			Reserved
1:0	vid_mode_type	R/W	This field indicates the video mode transmission type as follows: ■ 00: Non-burst with sync pulses ■ 01: Non-burst with sync events ■ 10 and 11: Burst mode

6.2.16 VID_PKT_SIZE

- **Description:** This register configures the video packet size.
- **Address offset:** 0x03C
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-17 Video Packet Size Register

Bits	Name	Access Type	Description
31:14			Reserved
13:0	vid_pkt_size	R/W	This field configures the number of pixels in a single video packet. 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.

6.2.17 VID_NUM_CHUNKS

- **Description:** This register configures the number of chunks to use. The data in each chunk has the size provided by VID_PKT_SIZE.
- **Address offset:** 0x040
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-18 Number Of Chunks Register

Bits	Name	Access Type	Description
31:13			Reserved
12:0	vid_num_chunks	R/W	This register configures the number of chunks to be transmitted during a Line period (a chunk consists of a video packet and a null packet). If set to 0 or 1, the video line is transmitted in a single packet. If set to 1, the packet is part of a chunk, so a null packet follows it if vid_null_size > 0. Otherwise, multiple chunks are used to transmit each video line.

6.2.18 VID_NULL_SIZE

- **Description:** This register configures the size of null packets.
- **Address offset:** 0x044
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-19 Null Packet Size Register

Bits	Name	Access Type	Description
31:13			Reserved
12:0	vid_null_size	R/W	This register configures the number of bytes inside a null packet. Setting it to 0 disables the null packets.

6.2.19 VID_HSA_TIME

- **Description:** This register configures the video Horizontal Synchronism Active (HSA) time.
- **Address offset:** 0x048
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-20 Horizontal Sync Active Time Register

Bits	Name	Access Type	Description
31:12			Reserved
11:0	vid_hsa_time	R/W	This field configures the Horizontal Synchronism Active period in lane byte clock cycles.

6.2.20 VID_HBP_TIME

- **Description:** This register configures the video Horizontal Back Porch (HBP) time.
- **Address offset:** 0x04C
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-21 Horizontal Back Porch Time Register

Bits	Name	Access Type	Description
31:12			Reserved
11:0	vid_hbp_time	R/W	This field configures the Horizontal Back Porch period in lane byte clock cycles.

6.2.21 VID_HLINE_TIME

- **Description:** This register configures the overall time for each video line.
- **Address offset:** 0x050
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-22 Line Time Register

Bits	Name	Access Type	Description
31:15			Reserved
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.

6.2.22 VID_VSA_LINES

- **Description:** This register configures the Vertical Synchronism Active (VSA) period.
- **Address offset:** 0x054
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-23 Vertical Sync Active Period Register

Bits	Name	Access Type	Description
31:10			Reserved
9:0	vsa_lines	R/W	This field configures the Vertical Synchronism Active period measured in number of horizontal lines.

6.2.23 VID_VBP_LINES

- **Description:** This register configures the Vertical Back Porch (VBP) period.
- **Address offset:** 0x058
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-24 Vertical Back Porch Period Register

Bits	Name	Access Type	Description
31:10			Reserved
9:0	vbp_lines	R/W	This field configures the Vertical Back Porch period measured in number of horizontal lines.

6.2.24 VID_VFP_LINES

- **Description:** This register configures the Vertical Front Porch (VFP) period.
- **Address offset:** 0x05C
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-25 Vertical Front Porch Period Register

Bits	Name	Access Type	Description
31:10			Reserved
9:0	vfp_lines	R/W	This field configures the Vertical Front Porch period measured in number of horizontal lines.

6.2.25 VID_VACTIVE_LINES

- **Description:** This register configures the vertical resolution of the video.
- **Address offset:** 0x060
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-26 Vertical Resolution Register

Bits	Name	Access Type	Description
31:14			Reserved
13:0	v_active_lines	R/W	This field configures the Vertical Active period measured in number of horizontal lines.

6.2.26 EDPI_CMD_SIZE

- **Description:** This register configures the size of the eDPI packets.
- **Address offset:** 0x064
- **Dependency:** DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-27 Edpi Packet Size Register

Bits	Name	Access Type	Description
31:16			Reserved
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.

6.2.27 CMD_MODE_CFG

- **Description:** This register configures several aspects of the command mode operation, tearing effect, acknowledge for each packet, and the speed mode to transmit each data type related to commands.
- **Address offset:** 0x068
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-28 Command Mode Configuration Register

Bits	Name	Access Type	Description
31:25			Reserved
24	max_rd_pkt_size	R/W	This bit configures the maximum read packet size command transmission type: ■ 0: High-speed ■ 1: Low-power
23:20			Reserved
19	dcs_lw_tx	R/W	This bit configures the DCS long write packet command transmission type: ■ 0: High-speed ■ 1: Low-power
18	dcs_sr_0p_tx	R/W	This bit configures the DCS short read packet with zero parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
17	dcs_sw_1p_tx	R/W	This bit configures the DCS short write packet with one parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
16	dcs_sw_0p_tx	R/W	This bit configures the DCS short write packet with zero parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
15			Reserved

Table 6-28 Command Mode Configuration Register (Continued)

Bits	Name	Access Type	Description
14	gen_lw_tx	R/W	This bit configures the Generic long write packet command transmission type: ■ 0: High-speed ■ 1: Low-power
13	gen_sr_2p_tx	R/W	This bit configures the Generic short read packet with two parameters command transmission type: ■ 0: High-speed ■ 1: Low-power
12	gen_sr_1p_tx	R/W	This bit configures the Generic short read packet with one parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
11	gen_sr_0p_tx	R/W	This bit configures the Generic short read packet with zero parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
10	gen_sw_2p_tx	R/W	This bit configures the Generic short write packet with two parameters command transmission type: ■ 0: High-speed ■ 1: Low-power
9	gen_sw_1p_tx	R/W	This bit configures the Generic short write packet with one parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
8	gen_sw_0p_tx	R/W	This bit configures the Generic short write packet with zero parameter command transmission type: ■ 0: High-speed ■ 1: Low-power
7:2			Reserved
1	ack_rqst_en	R/W	When set to 1, this bit enables the acknowledge request after each packet transmission.
0	tear_fx_en	R/W	When set to 1, this bit enables the tearing effect acknowledge request.

6.2.28 GEN_HDR

- **Description:** This register sets the header for new packets sent using the Generic interface.
- **Address offset:** 0x06C
- **Dependency:** DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-29 Generic Packet Header Configuration Register

Bits	Name	Access Type	Description
31:24			Reserved
23:16	gen_wc_msbyte	R/W	This field configures the most significant byte of the header packet's word count for long packets or data 1 for short packets.
15:8	gen_wc_lsbyte	R/W	This field configures the least significant byte of the header packet's Word count for long packets or data 0 for short packets.
7:6	gen_vc	R/W	This field configures the virtual channel id of the header packet.
5:0	gen_dt	R/W	This field configures the packet data type of the header packet.

6.2.29 GEN_PLD_DATA

- **Description:** This register sets the payload for the packets sent using the Generic interface and, when read, returns the contents of the read responses from the peripheral.
- **Address offset:** 0x070
- **Dependency:** DSI_GENERIC=1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-30 Generic Payload Data In And Out Register

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

6.2.30 CMD_PKT_STATUS

- **Description:** This register contains information about the status of the FIFOs related to the DBI and the Generic interfaces.
- **Address offset:** 0x074
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Access:** Read

Table 6-31 Command Packet Status Register

Bits	Name	Access Type	Description
31:15			Reserved
14	dbi_rd_cmd_busy	R	This bit is set when a read command is issued and cleared when the entire response is stored in the FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x0
13	dbi_pld_r_full	R	This bit indicates the full status of the DBI read payload FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x0
12	dbi_pld_r_empty	R	This bit indicates the empty status of the DBI read payload FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x1
11	dbi_pld_w_full	R	This bit indicates the full status of the DBI write payload FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x0
10	dbi_pld_w_empty	R	This bit indicates the empty status of the DBI write payload FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x1
9	dbi_cmd_full	R	This bit indicates the full status of the DBI command FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x0
8	dbi_cmd_empy	R	This bit indicates the empty status of the DBI command FIFO. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved. **Value after reset**: 0x1

Table 6-31 Command Packet Status Register (Continued)

Bits	Name	Access Type	Description
7			Reserved
6	gen_rd_cmd_busy	R	This bit is set when a read command is issued and cleared when the entire response is stored in the FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x0
5	gen_pld_r_full	R	This bit indicates the full status of the generic read payload FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x0
4	gen_pld_r_empty	R	This bit indicates the empty status of the generic read payload FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x1
3	gen_pld_w_full	R	This bit indicates the full status of the generic write payload FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x0
2	gen_pld_w_empty	R	This bit indicates the empty status of the generic write payload FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x1
1	gen_cmd_full	R	This bit indicates the full status of the generic command FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x0
0	gen_cmd_empty	R	This bit indicates the empty status of the generic command FIFO. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved. **Value after reset**: 0x1

6.2.31 TO_CNT_CFG

- **Description:** This register configures the counters that trigger the timeout errors. These are used to warn the system about a failure, through an interrupt, and restart the core in case of unexpected situations that cause deadlock conditions.
- **Address offset:** 0x078
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-32 Timeout Timers Configuration Register

Bits	Name	Access Type	Description
31:16	hstx_to_cnt	R/W	This field configures the timeout counter that triggers a high-speed transmission timeout contention detection (measured in TO_CLK_DIVISION cycles). If using the non-burst mode and there is no sufficient time to switch from HS to LP and back in the period which is from one line data finishing to the next line sync start, the DSI link returns the LP state once per frame, then you should configure the TO_CLK_DIVISION and hstx_to_cnt to be in accordance with: hstx_to_cnt * lanebyteclkperiod * TO_CLK_DIVISION >= the time of one FRAME data transmission * (1 + 10%) In burst mode, RGB pixel packets are time-compressed, leaving more time during a scan line. Therefore, if in burst mode and there is sufficient time to switch from HS to LP and back in the period of time from one line data finishing to the next line sync start, the DSI link can return LP mode and back in this time interval to save power. For this, configure the TO_CLK_DIVISION and hstx_to_cnt to be in accordance with: hstx_to_cnt * lanebyteclkperiod * TO_CLK_DIVISION >= the time of one LINE data transmission * (1 + 10%)
15:0	lprx_to_cnt	R/W	This field configures the timeout counter that triggers a low-power reception timeout contention detection (measured in TO_CLK_DIVISION cycles).

6.2.32 HS_RD_TO_CNT

- **Description:** This register configures the Peripheral Response timeout after high-speed read operations.
- **Address offset:** 0x07C
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-33 Peripheral Response Timeout Definition after High-Speed Read Operations Register

Bits	Name	Access Type	Description
31:16			Reserved
15:0	hs_rd_to_cnt	R/W	This field sets a period for which the DWC_mipi_dsi_host keeps the link still, after sending a high-speed read operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

6.2.33 LP_RD_TO_CNT

- **Description:** This register configures the Peripheral Response timeout after the low-power read operations.
- **Address offset:** 0x080
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-34 Peripheral Response Timeout Definition after Low-Power Read Operations Register

Bits	Name	Access Type	Description
31:16			Reserved
15:0	lp_rd_to_cnt	R/W	This field sets a period for which the DWC_mipi_dsi_host keeps the link still, after sending a low-power read operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

6.2.34 HS_WR_TO_CNT

- **Description:** This register configures the Peripheral Response timeout after the high-speed write operations.
- **Address offset:** 0x084
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-35 Peripheral Response Timeout Definition after High-Speed Write Operations Register

Bits	Name	Access Type	Description
31:25			Reserved
24	presp_to_mode	R/W	When set to 1, this bit ensures that the peripheral response timeout caused by hs_wr_to_cnt is used only once per eDPI frame, when both the following conditions are met: ■ 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. **Dependency**: DSI_DATAINTERFACE = 4. Otherwise, this bit is reserved.
23:16			Reserved
15:0	hs_wr_to_cnt	R/W	This field sets a period for which the DWC_mipi_dsi_host keeps the link inactive after sending a high-speed write operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

6.2.35 LP_WR_TO_CNT

- **Description:** This register configures the Peripheral Response timeout after the low-power write operations.
- **Address offset:** 0x088
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-36 Peripheral Response Timeout Definition after Low-Power Write Operations Register

Bits	Name	Access Type	Description
31:16			Reserved
15:0	lp_wr_to_cnt	R/W	This field sets a period for which the DWC_mipi_dsi_host keeps the link still, after sending a low-power write operation. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

6.2.36 BTA_TO_CNT

- **Description:** This register configures the Peripheral Response timeout after the Bus Turnaround completion.
- **Address offset:** 0x08C
- **Dependency:** DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-37 Peripheral Response Timeout Definition after Bus Turnaround Register

Bits	Name	Access Type	Description
31:16			Reserved
15:0	bta_to_cnt	R/W	This field sets a period for which the DWC_mipi_dsi_host keeps the link still, after completing a Bus Turn-Around. This period is measured in cycles of lanebyteclk. The counting starts when the D-PHY enters the Stop state and causes no interrupts.

6.2.37 SDF_3D

- **Description:** This register stores the 3D control information for the VSS packets in video mode.
- **Address offset:** 0x090
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-38 3D Control Configuration Register

Bits	Name	Access Type	Description
31:17			Reserved
16	send_3d_cfg	R/W	When set, causes the next VSS packet to include 3D control payload in every VSS packet.
15:6			Reserved
5	right_first	R/W	This bit defines the left or right order: ■ 0: Left eye data is sent first, and then the right eye data is sent. ■ 1: Right eye data is sent first, and then the left eye data is sent.
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: ■ 0: No sync pulses between left and right data ■ 1: Sync pulse (HSYNC, VSYNC, blanking) between left and right data
3:2	format_3d	R/W	This field defines the 3D image format: ■ 00: Line (alternating lines of left and right data) ■ 01: Frame (alternating frames of left and right data) ■ 10: Pixel (alternating pixels of left and right data) ■ 11: Reserved
1:0	mode_3d	R/W	This field defines the 3D mode on/off and display orientation: ■ 00: 3D mode off (2D mode on) ■ 01: 3D mode on, portrait orientation ■ 10: 3D mode on, landscape orientation ■ 11: Reserved

6.2.38 LPCLK_CTRL

- **Description:** This register configures using the non-continuous clock in the clock lane.
- **Address offset:** 0x094
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-39 Low-power in Clock Lane Register

Bits	Name	Access Type	Description
31:2			Reserved
1	auto_clklane_ctrl	R/W	This bit enables the automatic mechanism to stop providing clock in the clock lane when time allows.
0	phy_txrequestclkhs	R/W	This bit controls the D-PHY PPI txrequestclkhs signal.

6.2.39 PHY_TMR_LPCLK_CFG

- **Description:** This register sets the time that the DWC_mipi_dsi_host assumes in calculations for the clock lane to switch between high-speed and low-power.
- **Address offset:** 0x098
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-40 D-PHY Timing Configuration for the Clock Lane Register

Bits	Name	Access Type	Description
31:26			Reserved
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.
15:10			Reserved
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.

6.2.40 PHY_TMR_CFG

- **Description:** This register sets the time that the DWC_mipi_dsi_host assumes in calculations for the data lanes to switch between high-speed and low-power.
- **Address offset:** 0x09C
- **Dependency:** DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-41 D-PHY Data Lanes Timing Configuration Register

Bits	Name	Access Type	Description
31:24	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.
23:16	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.
15			Reserved
14:0	max_rd_time	R/W	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.

6.2.41 PHY_RSTZ

- **Description:** This register controls the resets and the PLL of the D-PHY.
- **Address offset:** 0x0A0
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-42 D-PHY Reset Control Register

Bits	Name	Access Type	Description
31:4			Reserved
3	phy_forcepll	R/W	When the D-PHY is in ULPS, this bit enables the D-PHY PLL. **Dependency**: DSI_HOST_FPGA = 0. Otherwise, this bit is reserved.
2	phy_enableclk	R/W	When set to1, this bit enables the D-PHY Clock Lane module.

Table 6-42 D-PHY Reset Control Register (Continued)

Bits	Name	Access Type	Description
1	phy_rstz	R/W	When set to 0, this bit places the digital section of the D-PHY in the reset state.
0	phy_shutdownz	R/W	When set to 0, this bit places the D-PHY macro in power-down state.

6.2.42 PHY_IF_CFG

- **Description:** This register configures the number of active lanes and the minimum time to remain in the Stop state.
- **Address offset:** 0x0A4
- **Value after reset:** DSI_HOST_NUMBER_OF_LANES-1
- **Access:** Read/Write

Table 6-43 D-PHY Interface Configuration Register

Bits	Name	Access Type	Description
31:16			Reserved
15:8	phy_stop_wait_time	R/W	This field configures the minimum wait period to request a high-speed transmission after the Stop state.
7:2			Reserved
1:0	n_lanes	R/W	This field configures the number of active data lanes: ■ 00: One data lane (lane 0) ■ 01: Two data lanes (lanes 0 and 1) ■ 10: Three data lanes (lanes 0, 1, and 2) ■ 11: Four data lanes (lanes 0, 1, 2, and 3)

6.2.43 PHY_ULPS_CTRL

- **Description:** This register configures entering and leaving the ULPS in the D-PHY.
- **Address offset:** 0x0A8
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-44 D-PHY Ultra Low-Power Control Register

Bits	Name	Access Type	Description
31:4			Reserved
3	phy_txexitulpslan	R/W	ULPS mode Exit on all active data lanes.
2	phy_txrequlpslan	R/W	ULPS mode Request on all active data lanes.
1	phy_txexitulpsclk	R/W	ULPS mode Exit on clock lane.
0	phy_txrequlpsclk	R/W	ULPS mode Request on clock lane.

6.2.44 PHY_TX_TRIGGERS

- **Description:** This register configures the signals that activate the triggers in the D-PHY.
- **Address offset:** 0x0AC
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-45 D-PHY Transmit Triggers Register

Bits	Name	Access Type	Description
31:4			Reserved
3:0	phy_tx_triggers	R/W	This field controls the trigger transmissions.

6.2.45 PHY_STATUS

- **Description:** This register contains information about the status of the D-PHY.
- **Address offset:** 0x0B0
- **Value after reset:** 0x00001540
- **Access:** Read

Table 6-46 D-PHY PPI Status Interface Register

Bits	Name	Access Type	Description
31:13			Reserved
12	phy_ulpsactivenot3lane	R	This bit indicates the status of ulpsactivenot3lane D-PHY signal. **Dependency**: DSI_HOST_NUMBER_OF_LANES > 3 If DSI_HOST_NUMBER_OF_LANES <= 3, this bit is reserved.
11	phy_stopstate3lane	R	This bit indicates the status of phystopstate3lane D-PHY signal. **Dependency**: DSI_HOST_NUMBER_OF_LANES > 3 If DSI_HOST_NUMBER_OF_LANES <= 3, this bit is reserved.
10	phy_ulpsactivenot2lane	R	This bit indicates the status of ulpsactivenot2lane D-PHY signal. **Dependency**: DSI_HOST_NUMBER_OF_LANES > 2 If DSI_HOST_NUMBER_OF_LANES <= 2, this bit is reserved.
9	phy_stopstate2lane	R	This bit indicates the status of phystopstate2lane D-PHY signal. **Dependency**: DSI_HOST_NUMBER_OF_LANES > 2 If DSI_HOST_NUMBER_OF_LANES <= 2, this bit is reserved.
8	phy_ulpsactivenot1lane	R	This bit indicates the status of ulpsactivenot1lane D-PHY signal. **Dependency**: DSI_HOST_NUMBER_OF_LANES > 1 If DSI_HOST_NUMBER_OF_LANES <= 1, this bit is reserved.
7	phy_stopstate1lane	R	This bit indicates the status of phystopstate1lane D-PHY signal. **Dependency**: DSI_HOST_NUMBER_OF_LANES > 1 If DSI_HOST_NUMBER_OF_LANES <= 1, this bit is reserved.
6	phy_rxulpsesc0lane	R	This bit indicates the status of rxulpsesc0lane D-PHY signal.
5	phy_ulpsactivenot0lane	R	This bit indicates the status of ulpsactivenot0lane D-PHY signal.
4	phy_stopstate0lane	R	This bit indicates the status of phystopstate0lane D-PHY signal.
3	phy_ulpsactivenotclk	R	This bit indicates the status of phyulpsactivenotclk D-PHY signal.
2	phy_stopstateclklane	R	This bit indicates the status of phystopstateclklane D-PHY signal.
1	phy_direction	R	This bit indicates the status of phydirection D-PHY signal.
0	phy_lock	R	This bit indicates the status of phylock D-PHY signal.

6.2.46 PHY_TST_CTRL0

- **Description:** This register controls the clock and the clear signals of the D-PHY vendor specific interface.
- **Address offset:** 0x0B4
- **Value after reset:** 0x00000001
- **Access:** Read/Write

Table 6-47 D-PHY Test Interface Control 0 Register

Bits	Name	Access Type	Description
31:2			Reserved
1	phy_testclk	R/W	This bit is used to clock the TESTDIN bus into the D-PHY.
0	phy_testclr	R/W	PHY test interface clear (active high).

6.2.47 PHY_TST_CTRL1

- **Description:** This register controls the data and the enable pins of the D-PHY vendor specific interface.
- **Address offset:** 0x0B8
- **Value after reset:** 0x00000000
- **Access:** Read/Write

Table 6-48 D-PHY Test Interface Control 1 Register

Bits	Name	Access Type	Description
31:17			Reserved
16	phy_testen	R/W	PHY test interface operation selector: ■ 1: The address write operation is set on the falling edge of the testclk signal. ■ 0: The data write operation is set on the rising edge of the testclk signal.
15:8	pht_testdout	R	PHY output 8-bit data bus for read-back and internal probing functionalities.
7:0	phy_testdin	R/W	PHY test interface input 8-bit data bus for internal register programming and test functionalities access.

6.2.48 INT_ST0

- **Description:** This register contains the status of the interrupt sources from the acknowledge reports and the D-PHY.
- **Address offset:** 0x0BC
- **Value after reset:** 0x00000000
- **Access:** Read

Table 6-49 Interrupt Status Register 0

Bits	Name	Access Type	Description
31:21			Reserved
20	dphy_errors_4	R	This bit indicates the LP1 contention error ErrContentionLP1 from Lane 0.
19	dphy_errors_3	R	This bit indicates the LP0 contention error ErrContentionLP0 from Lane 0.
18	dphy_errors_2	R	This bit indicates the ErrControl error from Lane 0.
17	dphy_errors_1	R	This bit indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0.
16	dphy_errors_0	R	This bit indicates ErrEsc escape entry error from Lane 0.
15	ack_with_err_15	R	This bit retrieves the DSI protocol violation from the Acknowledge error report.
14	ack_with_err_14	R	This bit retrieves the reserved (specific to device) from the Acknowledge error report.
13	ack_with_err_13	R	This bit retrieves the invalid transmission length from the Acknowledge error report.
12	ack_with_err_12	R	This bit retrieves the DSI VC ID Invalid from the Acknowledge error report.
11	ack_with_err_11	R	This bit retrieves the not recognized DSI data type from the Acknowledge error report.
10	ack_with_err_10	R	This bit retrieves the checksum error (long packet only) from the Acknowledge error report.
9	ack_with_err_9	R	This bit retrieves the ECC error, multi-bit (detected, not corrected) from the Acknowledge error report.
8	ack_with_err_8	R	This bit retrieves the ECC error, single-bit (detected and corrected) from the Acknowledge error report.
7	ack_with_err_7	R	This bit retrieves the reserved (specific to device) from the Acknowledge error report.

Table 6-49 Interrupt Status Register 0 (Continued)

Bits	Name	Access Type	Description
6	ack_with_err_6	R	This bit retrieves the False Control error from the Acknowledge error report.
5	ack_with_err_5	R	This bit retrieves the Peripheral Timeout error from the Acknowledge Error report.
4	ack_with_err_4	R	This bit retrieves the LP Transmit Sync error from the Acknowledge error report.
3	ack_with_err_3	R	This bit retrieves the Escape Mode Entry Command error from the Acknowledge error report.
2	ack_with_err_2	R	This bit retrieves the EoT Sync error from the Acknowledge error report.
1	ack_with_err_1	R	This bit retrieves the SoT Sync error from the Acknowledge error report.
0	ack_with_err_0	R	This bit retrieves the SoT error from the Acknowledge error report.

6.2.49 INT_ST1

- **Description:** This register contains the status of the interrupt sources related to timeouts, ECC, CRC, packet size, EoTp, Generic, and DBI interfaces.
- **Address offset:** 0x0C0
- **Value after reset:** 0x00000000
- **Access:** Read

Table 6-50 Interrupt Status Register 1

Bits	Name	Access Type	Description
31:18			Reserved
17	dbi_ilegal_comm_err	R	This bit indicates that an attempt to write an illegal command on the DBI interface is made and the core is blocked by transmission. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
16	dbi_pld_recv_err	R	This bit indicates that during a DBI read back packet, the payload FIFO becomes full and the received data is corrupted. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.

Table 6-50 Interrupt Status Register 1 (Continued)

Bits	Name	Access Type	Description
15	dbi_pld_rd_err	R	This bit indicates that during a DCS read data, the payload FIFO goes empty and the data sent to the interface is corrupted. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
14	dbi_pld_wr_err	R	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. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
13	dbi_cmd_wr_err	R	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. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
12	gen_pld_recev_err	R	This bit indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted. **Dependency**: DSI_GENERIC = 1 If DSI_GENERIC = 0, this bit is reserved.
11	gen_pld_rd_err	R	This bit indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved.
10	gen_pld_send_err	R	This bit indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved
9	gen_pld_wr_err	R	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. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved
8	gen_cmd_wr_err	R	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. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved
7	dpi_pld_wr_err	R	This bit indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted. **Dependency**: DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4. Otherwise, this bit is reserved.
6	eopt_err	R	This bit indicates that the EoTp packet is not received at the end of the incoming peripheral transmission.
5	pkt_size_err	R	This bit indicates that the packet size error is detected during the packet reception.

Table 6-50 Interrupt Status Register 1 (Continued)

Bits	Name	Access Type	Description
4	crc_err	R	This bit indicates that the CRC error is detected in the received packet payload. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1. Otherwise, this bit is reserved.
3	ecc_milti_err	R	This bit indicates that the ECC multiple error is detected in a received packet.
2	ecc_single_err	R	This bit indicates that the ECC single error is detected and corrected in a received packet.
1	to_lp_rx	R	This bit indicates that the low-power reception timeout counter reached the end and contention is detected.
0	to_hs_tx	R	This bit indicates that the high-speed transmission timeout counter reached the end and contention is detected.

6.2.50 INT_MSK0

- **Description:** This register configures the masks for the sources of interrupts that affect the INT_ST0 register.
- **Address offset:** 0x0C4
- **Dependency:** DSI_INTERRUPT_0_PIN = 0
- **Value after reset:** 0x001FFFFF
- **Access:** Read/Write

Table 6-51 Masks the Interrupt Generation Triggered by the ERROR_ST0 Register

Bits	Name	Access Type	Description
31:21			Reserved
20	dphy_errors_4	R/W	This bit indicates the LP1 contention error ErrContentionLP1 from Lane 0.
19	dphy_errors_3	R/W	This bit indicates the LP0 contention error ErrContentionLP0 from Lane 0.
18	dphy_errors_2	R/W	This bit indicates ErrControl control error from Lane 0.
17	dphy_errors_1	R/W	This bit indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0.
16	dphy_errors_0	R/W	This bit indicates ErrEsc escape entry error from Lane 0.

Table 6-51 Masks the Interrupt Generation Triggered by the ERROR_ST0 Register (Continued)

Bits	Name	Access Type	Description
15	ack_with_err_15	R/W	This bit retrieves the DSI protocol violation from the Acknowledge error report.
14	ack_with_err_14	R/W	This bit retrieves the reserved (specific to device) from the Acknowledge error report.
13	ack_with_err_13	R/W	This bit retrieves the invalid transmission length from the Acknowledge error report.
12	ack_with_err_12	R/W	This bit retrieves the DSI VC ID Invalid from the Acknowledge error report.
11	ack_with_err_11	R/W	This bit retrieves the not recognized DSI data type from the Acknowledge error report.
10	ack_with_err_10	R/W	This bit retrieves the checksum error (long packet only) from the Acknowledge error report.
9	ack_with_err_9	R/W	This bit retrieves the ECC error, multi-bit (detected, not corrected) from the Acknowledge error report.
8	ack_with_err_8	R/W	This bit retrieves the ECC error, single-bit (detected and corrected) from the Acknowledge error report.
7	ack_with_err_7	R/W	This bit retrieves the reserved (specific to device) from the Acknowledge error report.
6	ack_with_err_6	R/W	This bit retrieves the False Control error from the Acknowledge error report.
5	ack_with_err_5	R/W	This bit retrieves the Peripheral Timeout error from the Acknowledge error report.
4	ack_with_err_4	R/W	This bit retrieves the LP Transmit Sync error from the Acknowledge error report.
3	ack_with_err_3	R/W	This bit retrieves the Escape Mode Entry Command error from the Acknowledge error report.
2	ack_with_err_2	R/W	This bit retrieves the EoT Sync error from the Acknowledge error report.
1	ack_with_err_1	R/W	This bit retrieves the SoT Sync error from the Acknowledge error report.
0	ack_with_err_0	R/W	This bit retrieves the SoT error from the Acknowledge error report.

6.2.51 INT_MSK1

- **Description:** This register configures the masks for the sources of the interrupts that affect the INT_ST1 register.
- **Address offset:** 0x0C8
- **Dependency:** DSI_INTERRUPT_0_PIN = 0
- **Value after reset:** 0x0003FFFF
- **Access:** Read/Write

Table 6-52 Masks the Interrupt Generation Triggered by the ERROR_ST1 Register

Bits	Name	Access Type	Description
31:18			Reserved
17	dbi_ilegal_comm_err	R/W	This bit indicates that an attempt to write an illegal command on the DBI interface is made and the core is blocked by transmission. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
16	dbi_pld_recv_err	R/W	This bit indicates that during a DBI read back packet, the payload FIFO becomes full and the received data is corrupted. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
15	dbi_pld_rd_err	R/W	This bit indicates that during a DCS read data, the payload FIFO goes empty and the data sent to the interface is corrupted. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
14	dbi_pld_wr_err	R/W	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. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
13	dbi_cmd_wr_err	R/W	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. **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3. Otherwise, this bit is reserved.
12	gen_pld_recev_err	R/W	This bit indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved.
11	gen_pld_rd_err	R/W	This bit indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved.

Table 6-52 Masks the Interrupt Generation Triggered by the ERROR_ST1 Register (Continued)

Bits	Name	Access Type	Description
10	gen_pld_send_err	R/W	This bit indicates that during a Generic interface packet build, the payload FIFO becomes empty and corrupt data is sent. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved.
9	gen_pld_wr_err	R/W	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. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved.
8	gen_cmd_wr_err	R/W	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. **Dependency**: DSI_GENERIC = 1. Otherwise, this bit is reserved.
7	dpi_pld_wr_err	R/W	This bit indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted. **Dependency**: DSI_DATAINTERFACE = 2 or DSI_DATAINTERFACE = 3 or DSI_DATAINTERFACE = 4. Otherwise, this bit is reserved.
6	eopt_err	R/W	This bit indicates that the EoTp packet is not received at the end of the incoming peripheral transmission.
5	pkt_size_err	R/W	This bit indicates that the packet size error is detected during the packet reception.
4	crc_err	R/W	This bit indicates that the CRC error is detected in the received packet payload **Dependency**: DSI_DATAINTERFACE = 1 or DSI_DATAINTERFACE = 3 or DSI_GENERIC = 1. Otherwise, this bit is reserved.
3	ecc_milti_err	R/W	This bit indicates that the ECC multiple error is detected in a received packet.
2	ecc_single_err	R/W	This bit indicates that the ECC single error is detected and corrected in a received packet.
1	to_lp_rx	R/W	This bit indicates that the low-power reception timeout counter reached the end and contention is detected.
0	to_hs_tx	R/W	This bit indicates that the high-speed transmission timeout counter reached the end and contention is detected.

6.3 Static and Dynamic Registers

Some of the registers of the DWC_mipi_dsi_host are static and can be configured only when the DWC_mipi_dsi_host is idle and not transmitting. The dynamic registers can be configured when needed and do not require the DWC_mipi_dsi_host to be in reset or idle. Table 6-53 lists the static and dynamic registers.

Table 6-53 Static and Dynamic Registers

Static Read/Write Registers	Dynamic Read/Write Registers	Static Read-Only Registers	Dynamic Read-Only Registers
0x04 PWR_UP	0x34 MODE_CFG	0x00 VERSION	0x74 CMD_PKT_STATUS
0x08 CLKMGR_CFG	0x6C GEN_HDR		0xB0 PHY_STATUS
0x0C DPI_VCID	0x70 GEN_PLD_DATA		0xBC INT_ST0
0x1C DBI_VCID	0x90 SDF_3D		0xC0 INT_ST1
0x2C PCKHDL_CFG	0xA0 PHY_RSTZ		
0x3C VID_PKT_SIZE	0xA8 PHY_ULPS_CTRL		
0x4C VID_HBP_TIME	0xAC PHY_TX_TRIGGERS		
0x5C VID_VFP_LINES	0xB4 PHY_TST_CTRL0		
0x7C HS_RD_TO_CNT	0xB8 PHY_TST_CTRL1		
0x8C BTA_TO_CNT	0xC4 INT_MSK0		
0x9C PHY_TMR_CFG	0xC8 INT_MSK1		
0x10 DPI_COLOR_CODING			
0x14 DPI_CFG_POL			
0x18 DPI_LP_CMD_TIM			
0x20 DBI_CFG			
0x24 DBI_PARTITIONING_EN			
0x28 DBI_CMDSIZE			
0x30 GEN_VCID			
0x38 VID_MODE_CFG			
0x40 VID_NUM_CHUNKS			
0x44 VID_NULL_SIZE			
0x48 VID_HSA_TIME			
0x50 VID_HLINE_TIME			
0x54 VID_VSA_LINES			
0x58 VID_VBP_LINES			
0x60 VID_VACTIVE_LINES			
0x64 EDPI_CMD_SIZE			
0x68 CMD_MODE_CFG			
0x78 TO_CNT_CFG			
0x80 LP_RD_TO_CNT			
0x84 HS_WR_TO_CNT			
0x88 LP_WR_TO_CNT			
0x94 LPCLK_CTRL			
0x98 PHY_TMR_LPCLK_CFG			
0xA4 PHY_IF_CFG			

A
DBI Color Code Mapping Waveforms

This appendix contains the diagrams that show how the pixel-to-byte conversion is done for each color code mapping supported by the DBI Interface. The diagrams represent the cases where the interface works as type A with a fixed E, type A with clocked E, and type B.

Figure A-1 DSI 8-bit/8-bpp Byte Write

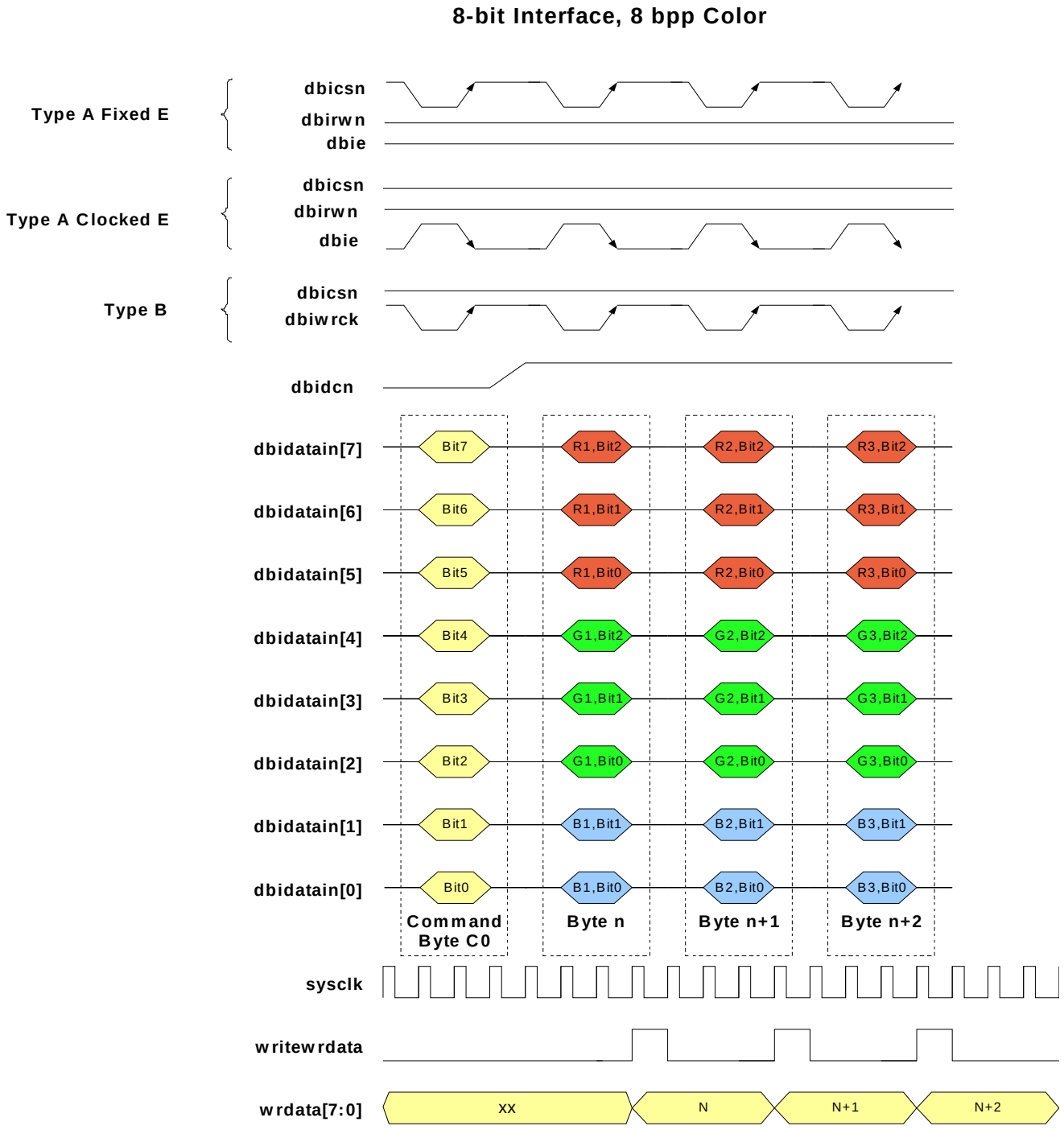


Figure A-2 DSI 8-bit/12-bpp Byte Write

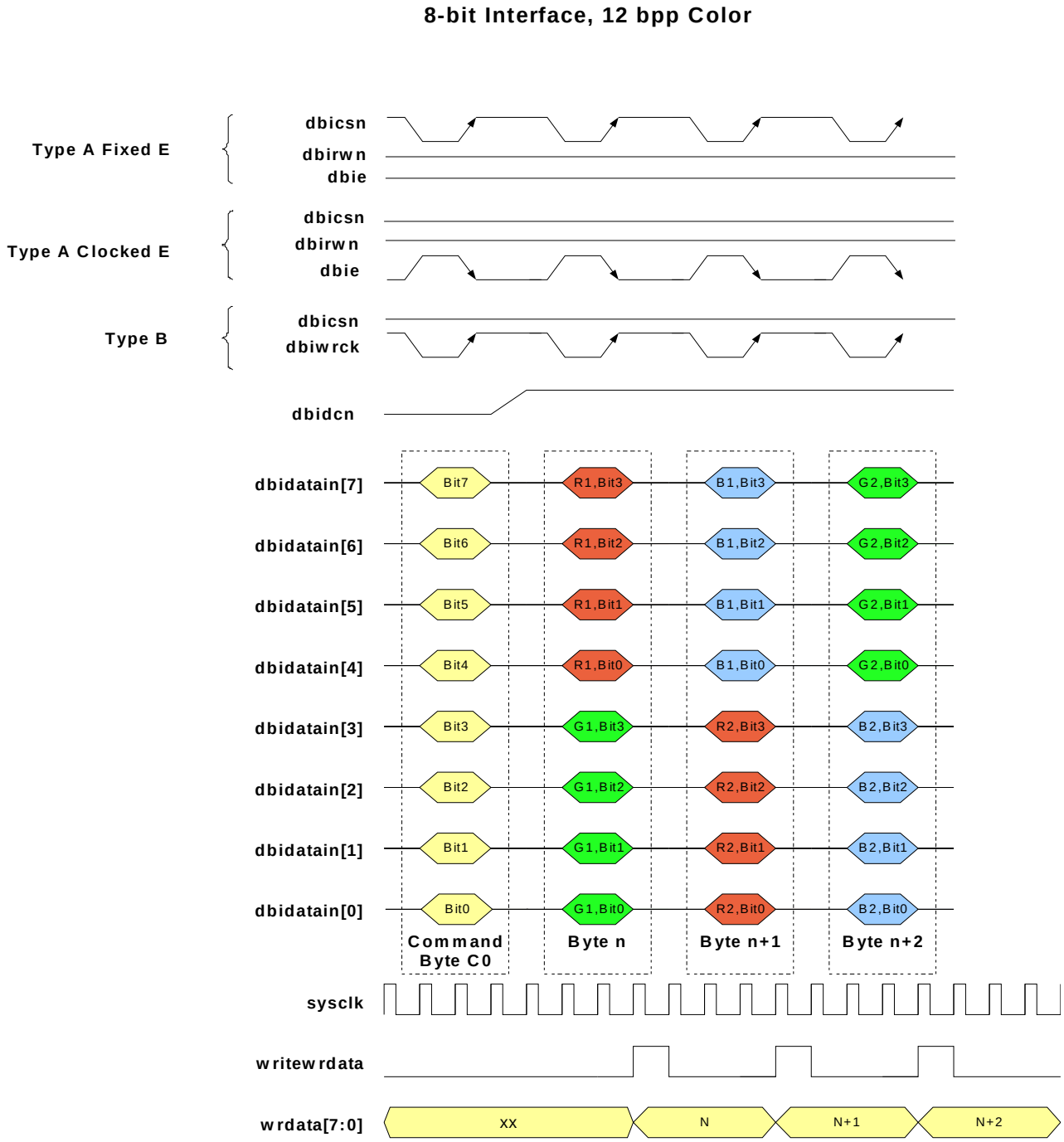


Figure A-3 DSI 8-bit/16-bpp Byte Write

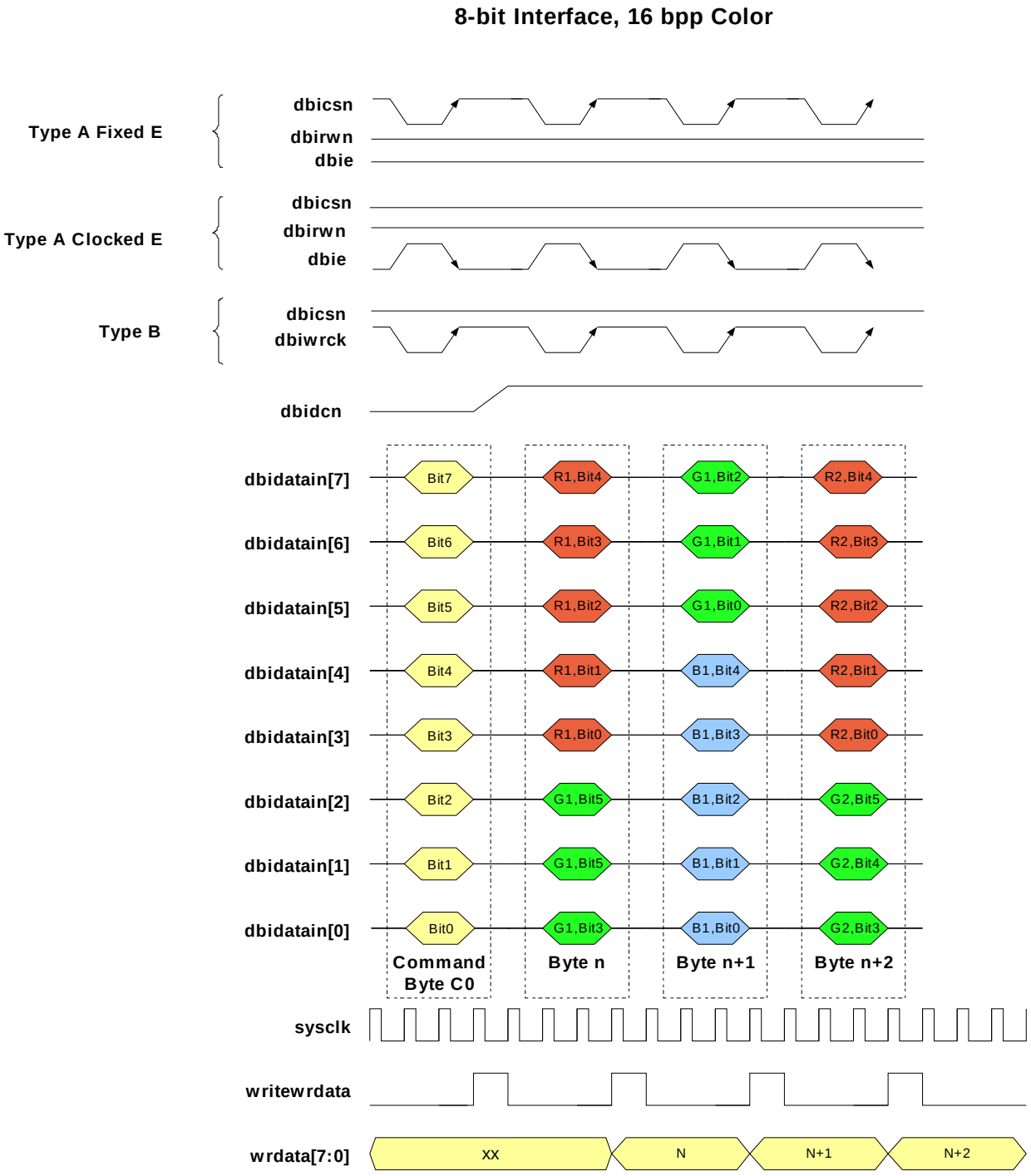


Figure A-4 DSI 8-bit/18-bpp Byte Write

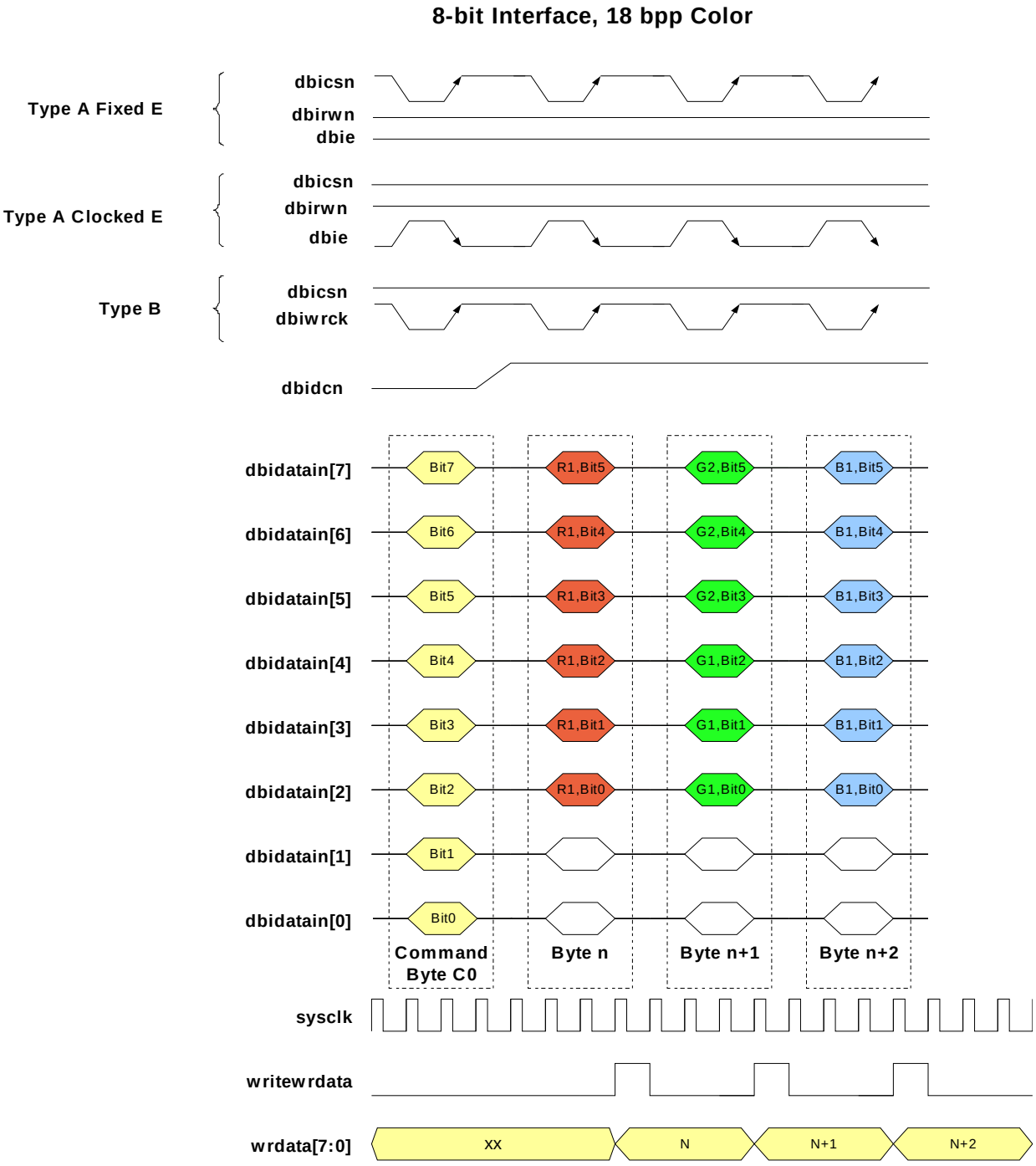


Figure A-5 DSI 8-bit/24-bpp Byte Write

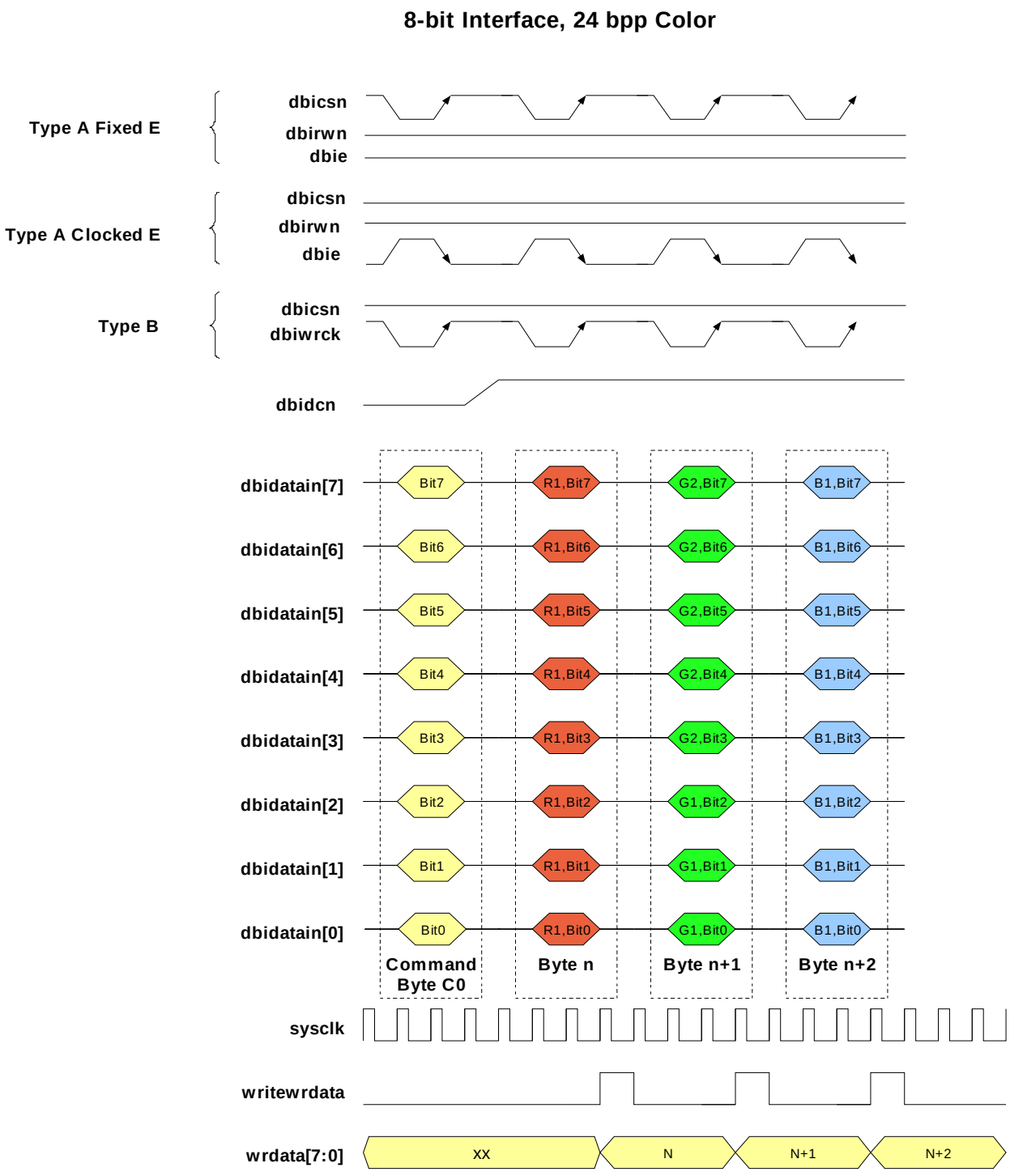


Figure A-6 DSI 9-bit/18-bpp Byte Write

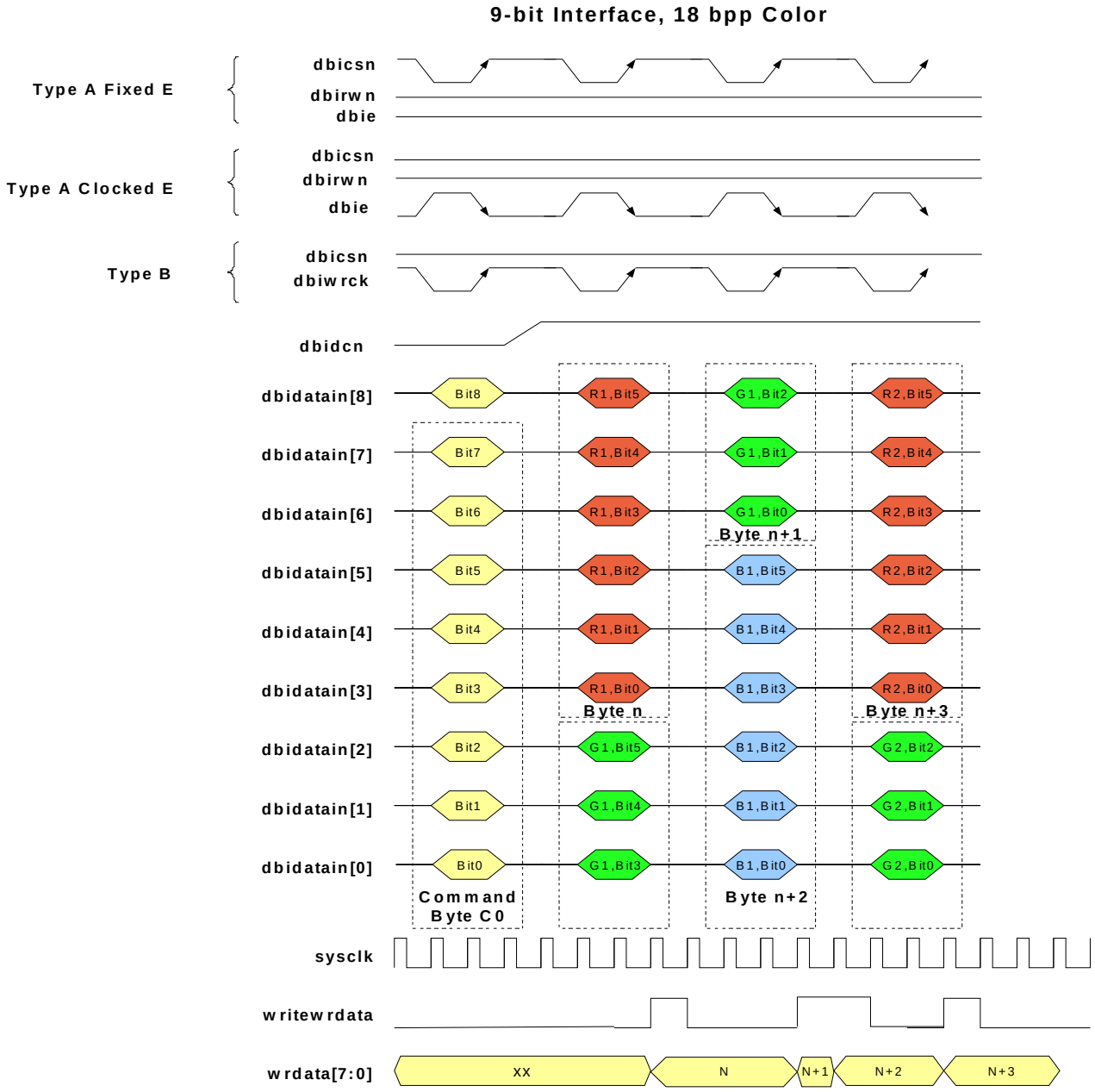


Figure A-7 DSI 16-bit/8-bpp Byte Write

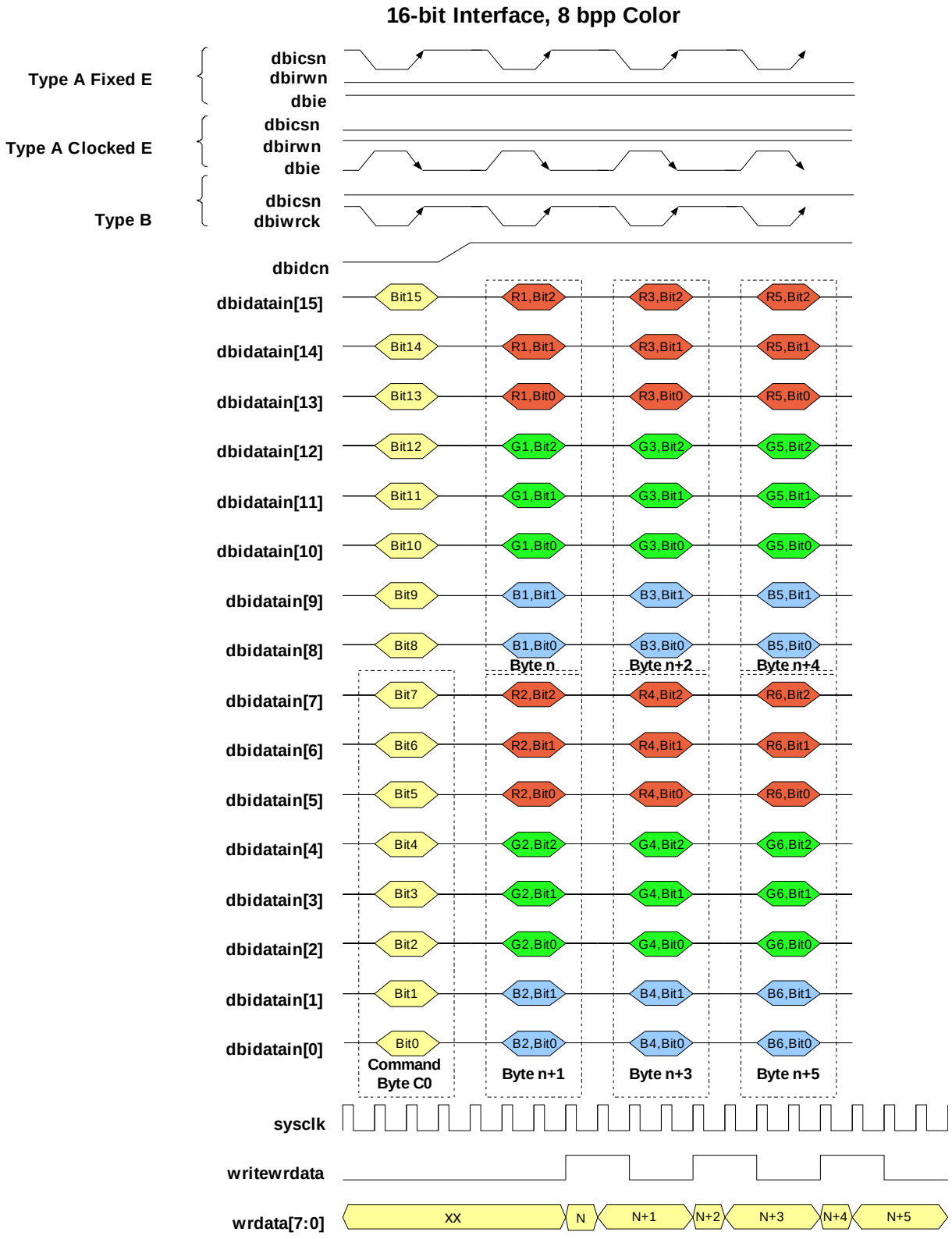


Figure A-8 DSI 16-bit/12-bpp Byte Write

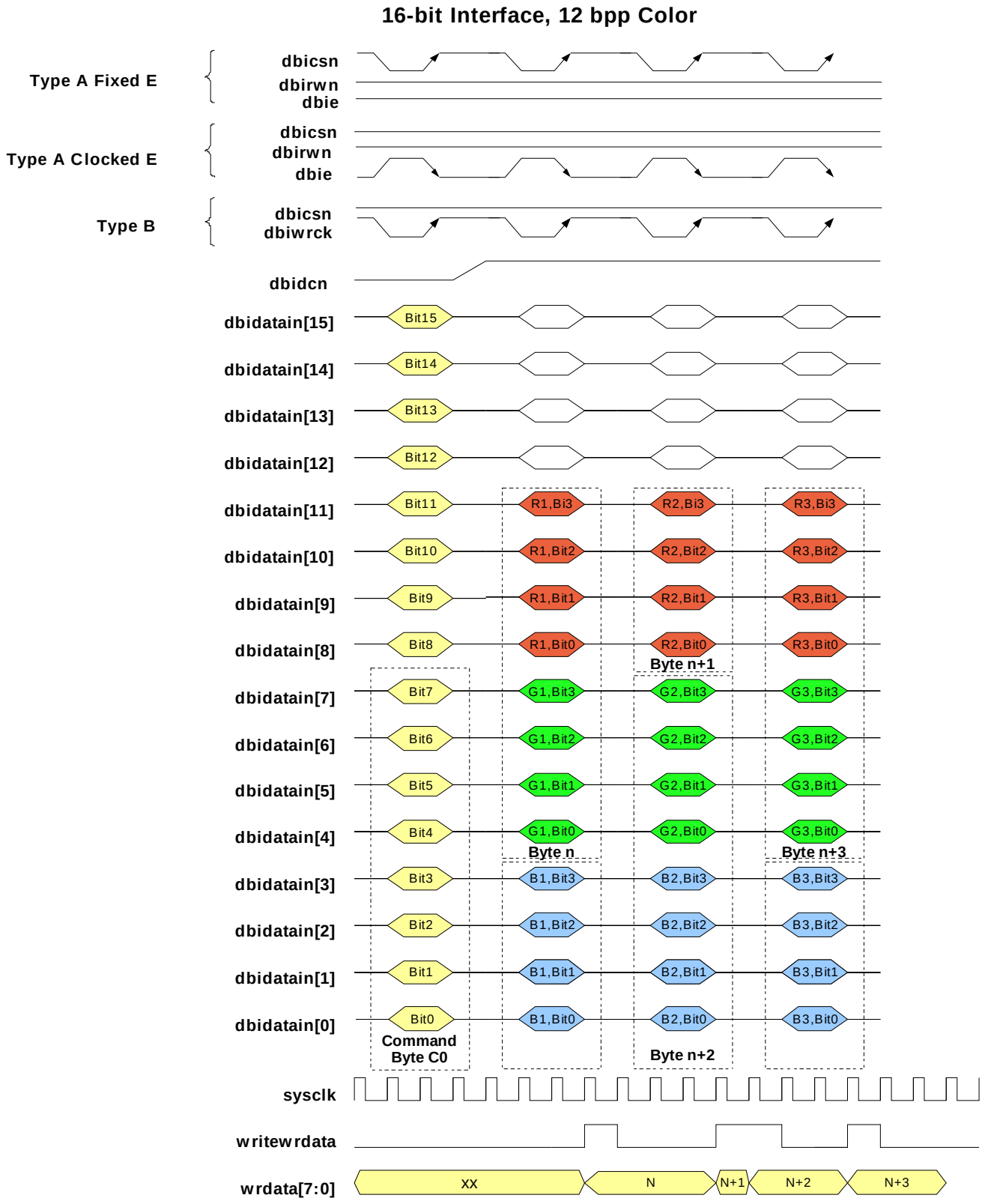


Figure A-9 DSI 16-bit/16-bpp Byte Write

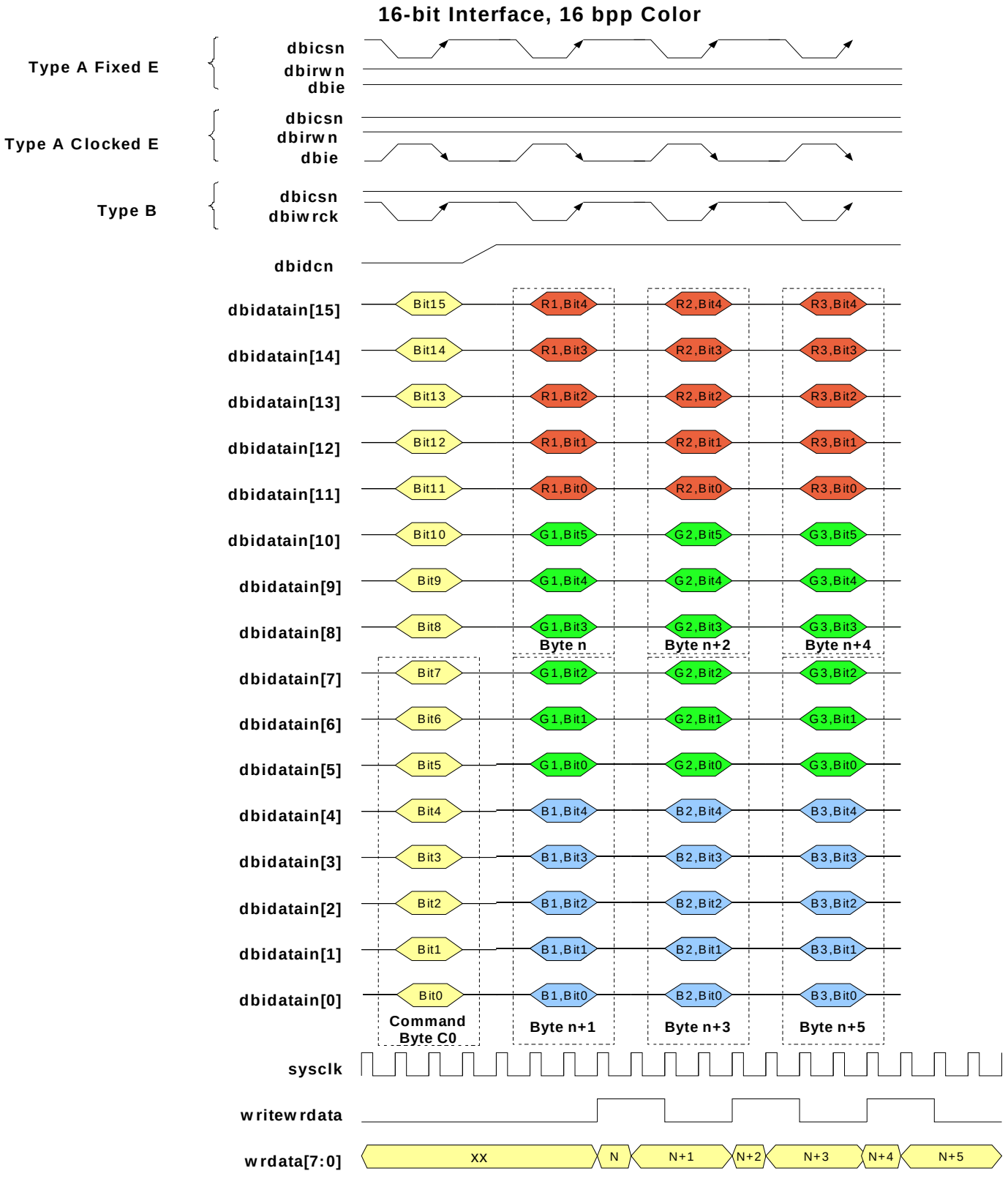


Figure A-10 DSI 16-bit/18-bpp Option 1 Byte Write

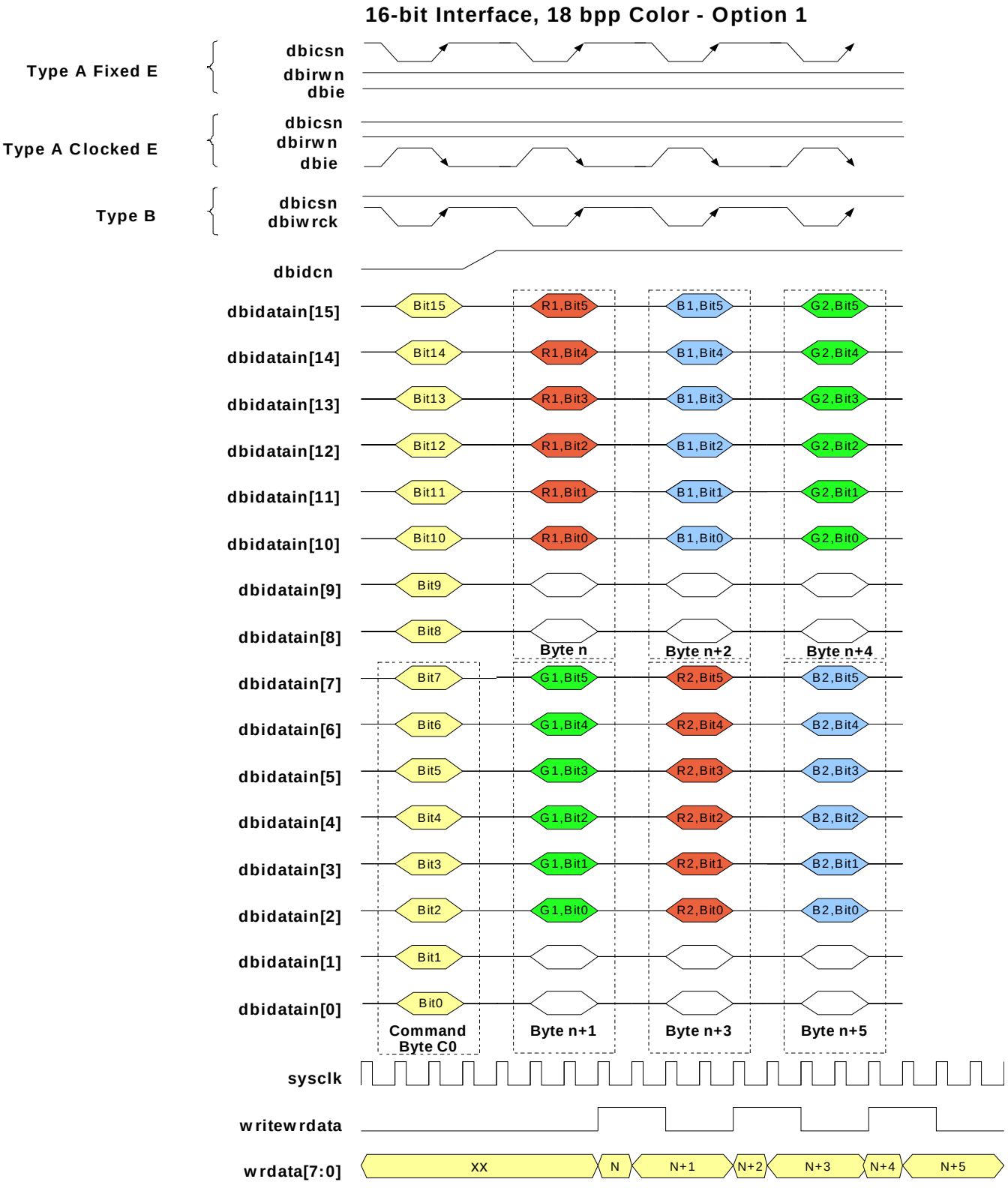


Figure A-11 DSI 16-bit/18-bpp Option 2 Byte Write

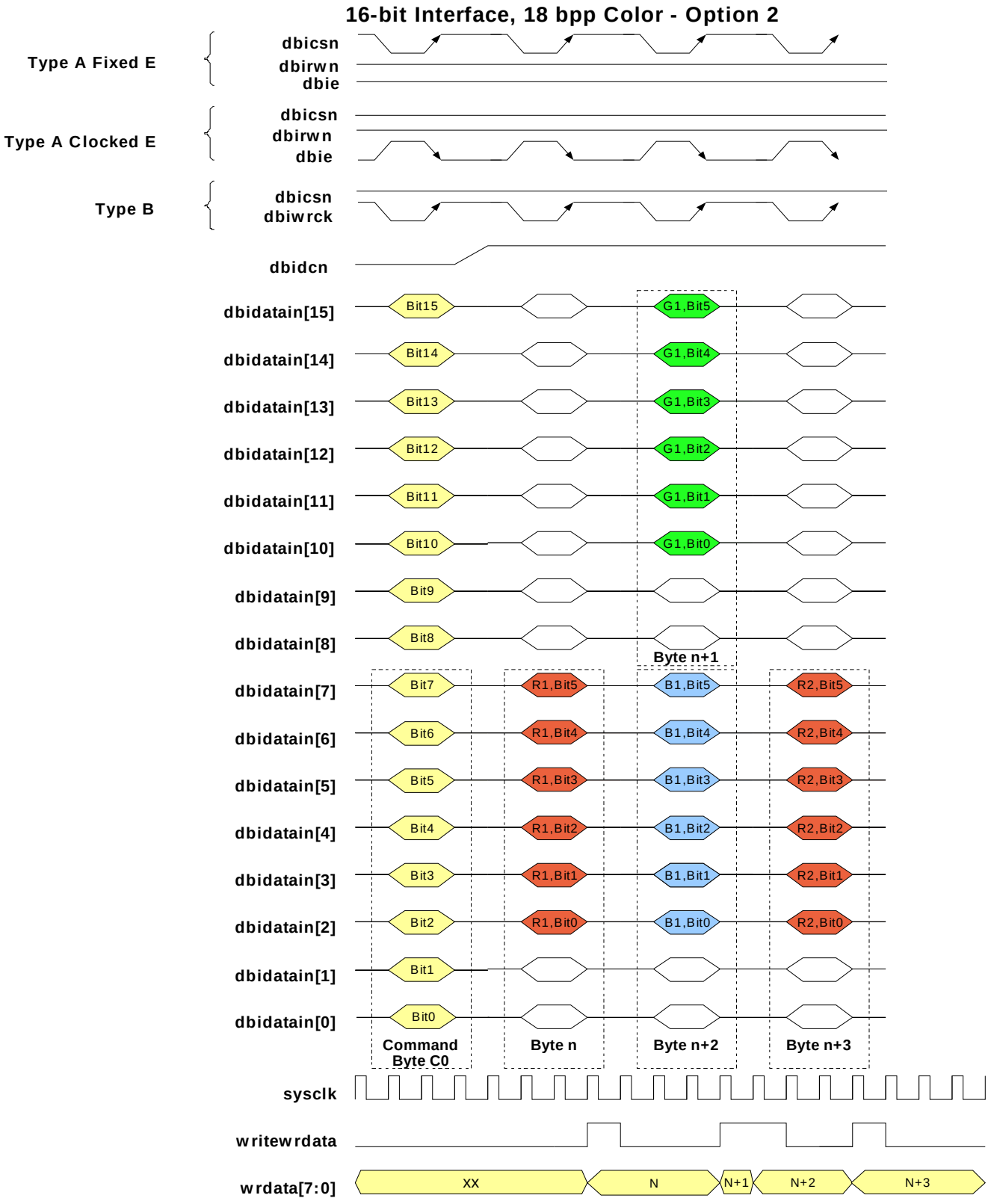


Figure A-12 DSI 16-bit/24-bpp Option 1 Byte Write

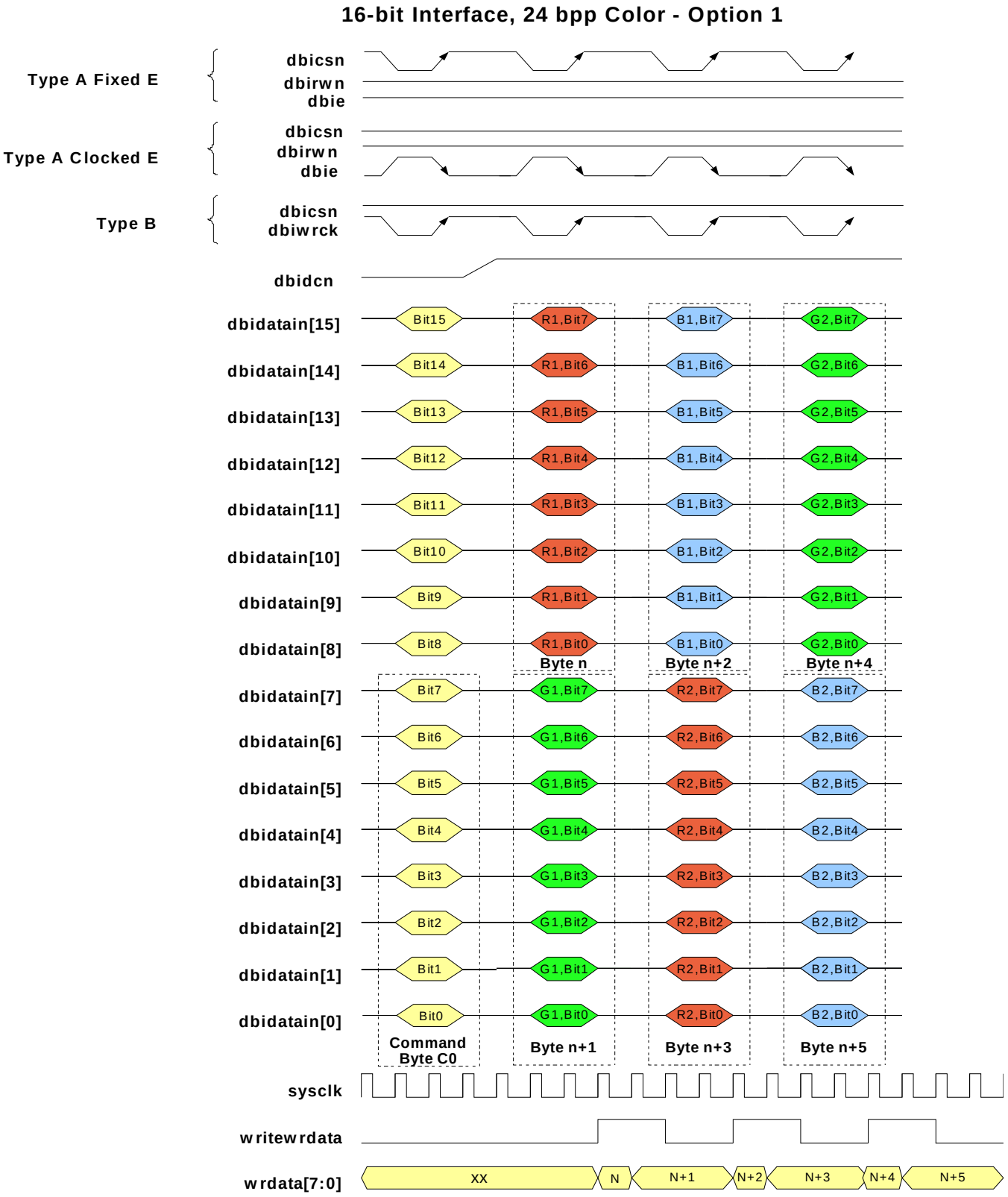
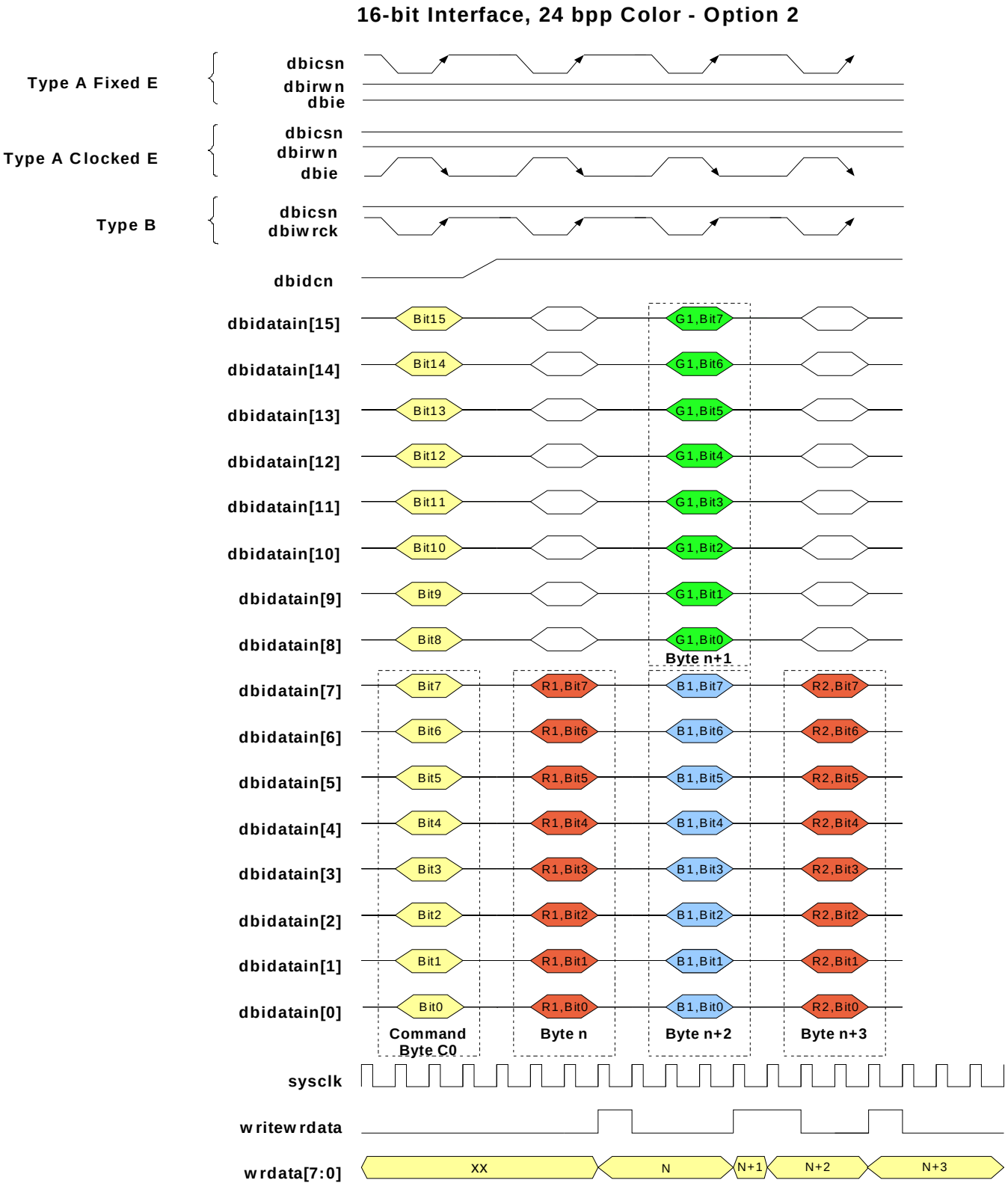


Figure A-13 DSI 16-bit/24-bpp Option 2 Byte Write



B
Error Handling

This appendix explains the conditions that trigger the interrupts and provides suggestions to recover from these error states.

B.1 Causes of Errors and Recovery

The interrupts triggered by the ERROR_ST0 or ERROR_ST1 register bits are error conditions and can occur only at the debug stage. Table B-1 explains the reasons that set off these interrupts and also explains how to recover from these interrupts.

Note 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.

In addition to the causes and suggestions for recovery provided in Table B-1, there can be other causes and methods of handling each interrupts.

Table B-1 Error cause and recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
ERROR_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 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 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 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 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 D-PHY Host recovers but the system should repeat the RX reception.

Table B-1 Error cause and recovery

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.	Refer to 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.	Refer to 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 is multiple errors present in the packet header (more than 2), so 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.

Table B-1 Error cause and recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
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.
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. Refer to the 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.	The device detects one of the following: ■ The LP-10 (LP request) is not followed by the remainder of a valid escape or turnaround sequence. ■ The LP-01 (HS request) is not followed by a bridge state (LP-00). The D-PHY communications are corrupted. This error is unrecoverable. Reset the DWC_mipi_dsi_host and the D-PHY. Refer to the section 7.1.6 of the DSI Specification 1.1.
5	ack_with_err_5	The Acknowledge with Error packet contains this error. The display timeout counters for a HS reception and LP transmission expire.	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. 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.
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.

Table B-1 Error cause and recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
2	ack_with_err_2	The Acknowledge with Error packet contains this error. The Device detects the HS 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.
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 was important, transmit the packets again. For additional information about this error, see section 7.1.1 of the DSI Specification 1.1.
ERROR_ST0 Register Bits			
17	dbi_illegal_comm_err	A command that is not defined in the DCS specification is driven in the DBI interface.	The system should check the DBI usage procedures. Commands that are not defined in the DCS Specification cannot be driven through the DBI interface.
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 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 and repeat the read procedure. Ensure that the read procedure is completed before reading the data through the DBI interface.
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 Table 2-1. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host and repeat the write procedure.

Table B-1 Error cause and recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
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, see Table 2-1. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host and repeat the write procedure.
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 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 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 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 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, see Table 2-1. Data stored in the FIFOs is corrupted. Reset the DWC_mipi_dsi_host and repeat the write procedure.

Table B-1 Error cause and recovery

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
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 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 by boundaries.	The integrity of the received data cannot be guaranteed. Reset the DWC_mipi_dsi_host and repeat the read procedure.
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 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 ignores all the following packets. The DWC_mipi_dsi_host should 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 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 automatically resets the controller side 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 automatically resets the controller side 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.