

AHEAD OF WHAT'S POSSIBLE™

GMSL2 User Guide

GMSL SerDes Applications Team

Version 26

May 5th, 2022

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

1 Table of Contents

GMSL2 User Guide 1
1 Table of Contents 2
Serial Link Features 4
2 GMSL2 Link Basics 5
3 Spread-Spectrum Clocking (SSC) 30
4 GMSL1 Legacy Mode (Backward Compatibility) 34
5 Power Manager and Sleep Mode 35
6 Clocks and Frequency Reference 36
7 PRBS Testing 51
8 Link Integrity Tools 58
9 Software Algorithms for Link Integrity Tools 59
Video Transmission 60
10 Video Basics 61
11 Camera Basics 68
12 Video Pipes 69
13 Dual-View 75
14 Display Stream Compression (DSC) 91
15 Forward Error Correction (FEC) 103
16 HDCP 112
17 Watermarking 113
18 Video Timing Generator (VTG) and Video Pattern Generator (VPG) 123
19 Video Crossbar 137
20 Color Lookup Table (LUT) 145
21 Frame Sync 149
22 Video Sync Pulse Outputs 159
Bidirectional Channels 165
23 Audio 166
24 I2C/UART 188
25 Touch-Screen Applications 227
26 I/O Matrix (MFP) 238
27 General-Purpose Input and Output (GPIO) 239
28 Serial Peripheral Interface (SPI) 252
29 RMII/RGMII (Ethernet) 267

Analog Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
ADI Confidential for Newvision under NDA

Bandwidth Calculations 295
30 GMSL2 Link System Bandwidth 296
Functional Safety 332
31 Functional Safety Introduction 333
32 ASIL 334
33 GMSL2 Error Reporting (ERRB Pin) 335
34 CRC Error Detection and ARQ Error Correction 365
35 Voltage Monitoring 372
36 ADC 378
37 Line Fault 379
38 Error Generator (ERRG) 384
Device Families 387
39 HDMI Serializers 388
40 DSI Serializers 402
41 Dual OLDI 450
42 eDP/DP Deserializers 465
43 CSI-2 Serializers and Deserializers 521
44 CSI-2 Quad Deserializers 609
45 MIPI D-PHY Deskew 717
Appendix 736
46 Revision History 736
47 Figures 740
48 Tables 745

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

Serial Link Features

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

2 GMSL2 Link Basics

2.1 GMSL2 Overview

This section provides a brief introduction to the proprietary physical interface (PHY) layer and link protocol used by GMSL2 serial links. Reference the GMSL2 Hardware Design Guide for a more detailed discussion of the PHY and the implementation of the Channel Specification.

GMSL2 serial links uses packet-based, bidirectional architecture with forward and reverse channels. The *forward channel* transfers data from the serializer to the deserializer; the *reverse channel* transfers data from the deserializer to the serializer. The programmable forward and reverse channel link rates are fixed and independent of the video pixel clock (PCLK). Typical GMSL2 devices have a forward serial bit rate of 3Gbps or 6Gbps and reverse channel serial bit rate of 187.5Mbps or 1.5Gbps (refer to Table 1 below for supported data rates by part number).

Note: *Transmit side* refers to the device in the serial link transmitting to the *receive side*. These arrangements are dependent upon application.

2.1.1 Serial Link Rates by Part Number

Table 1. GMSL2 Serial Link Rates by Part Number

Device Family	Part Number	GMSL2 Forward Channel Link Rates Supported		GMSL2 Reverse Channel Link Rates Supported	
		6Gbps	3Gbps	187Mbps	1.5Gbps
HDMI Serializers	MAX96751	X	X	X	
	MAX96751F	X	X	X	
	MAX96753	X	X	X	
	MAX96753F	X	X	X	
	MAX96751H	X	X	X	
Advanced HDMI Serializers	MAX96763	X	X	X	
	MAX96765	X	X	X	
	MAX96759	X	X	X	X
	MAX96761	X	X	X	X
	MAX96785	X	X	X	
	MAX96787	X	X	X	
DSI Serializers	MAX96755	X	X	X	
	MAX96755F	X	X	X	
	MAX96755H	X	X	X	
	MAX96757F	X	X	X	
	MAX96757	X	X	X	
	MAX9283	N/A	N/A	N/A	N/A
	MAX9285	N/A	N/A	N/A	N/A
Advanced DSI Serializers	MAX96789	X	X	X	
	MAX96789F	X	X	X	

ADI Confidential for New Revision under NDA
Maxim Integrated Confidential, Distribution Not Permitted
Provided to GMSL2 Customers Under NDA
Company Confidential,

	MAX96791	X	X	X	
	MAX96791F	X	X	X	
CSI-2 Serializers	MAX9295A	X	X	X	
	MAX9295B	X	X	X	
Advanced CSI-2 Serializers	MAX9295D	X	X	X	
	MAX9295E	X	X	X	
CSI-2 Camera Deserializers	MAX96784	X	X	X	X
	MAX9296A	X	X	X	
	MAX9296B	X	X	X	
CSI-2 Display Deserializers	MAX96756	X	X	X	
	MAX96756F	X	X	X	
	MAX96758F	X	X	X	
	MAX96758	X	X	X	
OLDI Deserializers	MAX96752	X	X	X	
	MAX96752F	X	X	X	
	MAX96754	X	X	X	
	MAX96754F	X	X	X	
eDP/DP Deserializers	MAX96772	X	X	X	
	MAX96774	X	X	X	
	MAX96776	X	X	X	X
	MAX96778	X	X	X	X
CSI-2 Quad Deserializers	MAX96712	X	X	X	
	MAX96722		X	X	

ADI Confidential for NewVision under NDA
Analog Confidential Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

2.2 GMSL2 Link Configurations

The GMSL2 architecture can support the following link configurations. Reference Table 4 for specific configuration options for each part.

2.2.1 Single Link Mode

In single link mode, a single serializer PHY connects to a single deserializer PHY (Figure 1). The available bandwidth is equal to the forward and reverse channel link rate that has been selected. In parts that only have a single GMSL2 PHY, this is the only link mode available.

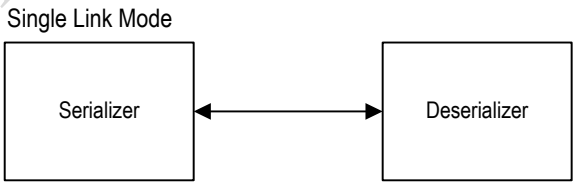


Figure 1. GMSL2 Single Link Mode

2.2.2 Splitter Mode

In splitter mode, a single serializer, via two GMSL2 PHYs, connects to two deserializers (Figure 2). The available bandwidth to each deserializer is equal to the forward and reverse channel link rate that has been selected.

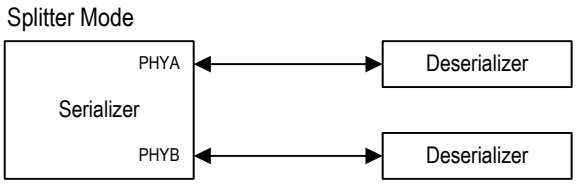


Figure 2. GMSL2 Splitter Mode

2.2.3 Reverse Splitter Mode

In reverse splitter mode, two serializers connect to a single deserializer via two GMSL2 PHYs (Figure 3). The available bandwidth into the deserializer is equal to the sum of the forward channel link rates that have been selected in each serializer, allowing up to 12Gbps of bandwidth to be aggregated into the deserializer.

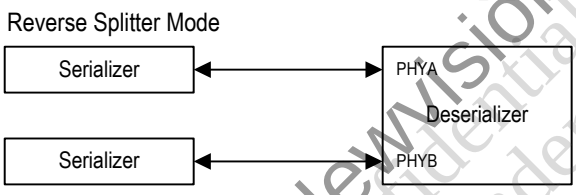


Figure 3. Reverse Splitter Mode

ADI Confidential for New Version under NDA
Analog Devices Confidential, Subject to Change under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

2.2.4 Dual Link Mode

In dual link mode, both PHYs of a single serializer connect to both PHYs of a single deserializer (Figure 4). The available bandwidth is equal to two times the forward and reverse channel link rate that has been selected. Both devices must support dual-link and PHY A in the serializer must connect to PHY A in the deserializer and likewise with PHY B. When using 6Gbps forward link rate, the maximum PCLK supported is 430MHz in dual-link mode. HDCP is supported in dual-link mode, but DSC is **not** supported.

2.2.4.1 Shielded-twisted-quad (STQ)

STQ cable can be used for dual-link mode applications. Up to 5m of Leoni Dacar® 636 cable can be used when operating in 6Gbps forward and 187.5Mbps reverse rate operation. No other make/model of STQ cable has been verified for dual-link operation. ‘

2.2.4.2 Coaxial Cables

Coaxial cables can be used for dual-link operation. The standard GMSL2 link requirements described in the GMSL2 Channel Specification must be adhered to when using coaxial cables in this application – with the addition of a cable-to-cable skew limit. When two coaxial cables are used for dual-link operation they must be relatively close in length to ensure PHY-to-PHY skew does not exceed 10ns. This means there should be no more than 2m difference in length.

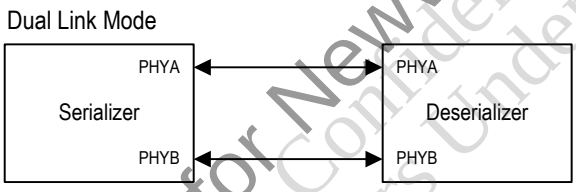


Figure 4. Dual Link Mode

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Copyright Maxim Integrated
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

2.3 *GMSL2 Link Rate and Configuration Programming*

The link rate is configured with the TX_RATE[1:0] and RX_RATE[1:0] registers in each device. See Table 2 below.

2.3.1 Programming Link Rate

Table 2. GMSL2 Link Rate Configuration

Link Rate	Serializer Registers	Deserializer Registers
Forward Channel Link Rate		
6Gbps	TX_RATE[1:0] = 10	RX_RATE[1:0] = 10
3Gbps	TX_RATE[1:0] = 01	RX_RATE[1:0] = 01
Reverse Channel Link Rate		
1.5Gbps	RX_RATE[1:0] = 11	TX_RATE[1:0] = 11
187Mbps	RX_RATE[1:0] = 00	TX_RATE[1:0] = 00

2.3.2 Programming Link Configuration

Link configuration is controlled with the AUTO_LINK and LINK_CFG[1:0] registers. After link configuration settings are changed, a RESET_ONESHOT command must be sent to either the serializer or deserializer for the link change to take effect. See the see *Resets* section for more information.

2.3.2.1 Auto Link Mode

By default, automatic link configuration mode is enabled (AUTO_LINK = 1). This means the device will attempt to lock in single link mode. It will automatically detect if PHYA or PHYB is connected to the remote device and will enable that PHY. If both PHYA and PHYB are connected to remote devices, it will connect in single link mode to whichever device is first detected.

Note: In auto link mode, the LINK_CFG[1:0] setting is ignored.

2.3.2.2 Manual Link Mode

Manual link mode should be used to manually select which PHY is used in single link mode and to configure splitter, reverse splitter, or dual link modes. To enable manual link mode, set AUTO_LINK = 0. See Table 3 for configuration details for each mode. Manual link mode can impact the time to achieve GMSL link lock. To optimize the link lock time, see *GMSL2 Link Lock Time Optimization* where the t_{RD} needs to be controlled. For typical single link applications, AUTO_LINK = 1 is recommended to simplify system level design.

Table 3. Manual Link Mode Configuration

Link Mode	Serializer Settings	Deserializer Settings
Single Link Mode (PHYA)	AUTO_LINK = 0 LINK_CFG[1:0] = 01	AUTO_LINK = 0 LINK_CFG[1:0] = 01
Single Link Mode (PHYB)	AUTO_LINK = 0 LINK_CFG[1:0] = 10	AUTO_LINK = 0 LINK_CFG[1:0] = 10

Analog Confidential for New Version under NDA
Maxim Confidential Distribute Only
Provided to GMSL2 Customer Company Confidential, Under NDA

Splitter Mode	AUTO_LINK = 0 LINK_CFG[1:0] = 11	In both deserializers, either use auto link mode or use manual mode and configure to single link mode for the connected PHY (auto link mode recommended)
Reverse Splitter Mode	In both serializers, either use auto link mode or use manual mode and configure to single link mode for the connected PHY (auto link mode recommended)	AUTO_LINK = 0 LINK_CFG[1:0] = 11
Dual Link Mode	AUTO_LINK = 0 LINK_CFG[1:0] = 00	AUTO_LINK = 0 LINK_CFG[1:0] = 00

Note: Link rate and configuration changes require a link reset to take effect. Write RESET_ONESHOT to 1 in the serializer after making changes.

2.3.2.3 Programming Example

```
# Set both serializer and deserializer to 6G/187M rate
0x80,0x0001,0x48
0x90,0x0001,0x02
# Set both parts to dual-link mode
0x80,0x0010,0x00
0x90,0x0010,0x00
# Write one-shot reset in serializer
0x80,0x0010,0x20
```

2.3.3 Standard and Glitchless Splitter Mode

There are two different implementations of splitter mode in GMSL devices: (standard) splitter mode and glitchless splitter mode. Standard splitter mode is supported by HDMI serializers and DSI serializers; glitchless splitter mode is supported by Advanced HDMI serializers and Advanced DSI serializers. Glitchless splitter mode enables the link to switch “glitchlessly” between splitter and single link modes by allowing the serializer to maintain lock even if only one deserializer is connected.

In standard splitter mode, the serializer must be connected to both deserializers to lock the GMSL2 link and send valid video streams and data from the communication channels. If one deserializer is disconnected or powered down, the GMSL2 link lock will be lost to both deserializers, and the LOCK pin on all three GMSL2 devices will go low. Splitter mode expects both links to be locked to transmit any data. To send data to a single deserializer, this restriction requires that the serializer be switched from splitter mode to single link mode.

The sequences below apply to devices that are in standard splitter mode and do not support glitchless splitter mode. For devices that support glitchless splitter mode, the switch from splitter to single link mode happens automatically without glitches to the video or sideband channels.

2.3.3.1 Standard Splitter Mode: Switch from Splitter Mode to Single Link Mode

An individual PHY in the serializer is programmed to single link mode. Enable single link mode by writing register 0x0010 with one of the following values:

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute Only Under NDA
Maxim Integrated Confidential, Distribute Only Under NDA
Provided to GMSL2_Customers Under NDA

- PHY A: 0x21
- PHY B: 0x22

This write will switch the link to single link mode and performs a “Oneshot Reset”. The following process takes <50ms to complete:

1. Reset the link into single link mode.
2. Self-clear the reset bit.
3. Recalibrate the canceller.
4. Auto-adapt the equalizer.
5. Link up to the deserializer.
6. Pull the LOCK pin high.

Alternatively, the serializer can be programmed to “Auto Link Mode”. In auto link mode, the serializer will automatically detect which PHY is connected to the deserializer and will enable that PHY. Program auto link mode by writing the serializer register 0x0010 to 0x30. The following process is automatically performed:

- Autodetect which of the two PHYs is connected to a deserializer.
- Reset the link to single link mode to the detected deserializer device.
- Self-clear the reset bit.
- Recalibrate the canceller.
- Auto-adapt the equalizer.
- Link up to the deserializer.
- Pull the LOCK pin high.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution Only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

2.3.3.2 Standard Splitter Mode: Switch from Single Link Mode to Splitter Mode

If both deserializers connected to the serializers are powered-up again, the serializer will remain linked to only one deserializer in single link mode until splitter mode is re-enabled. To re-enable splitter mode, write serializer register 0x0010 to 0x23. The following re-enabling process takes <50ms to complete:

- Reset the link to splitter mode to both deserializers.
- Self-clear the reset bit.
- Recalibrate the canceller.
- Auto-adapt the equalizer.
- Link up to both deserializers.
- Pull the LOCK pin high on all three devices.

2.3.4 GMSL1 Splitter Mode Configuration

Advanced DSI GMSL2 serializers can support up to two GMSL1 links in backward compatible splitter mode. This application may require additional configuration if the main control channel is in I2C mode (UART mode is not affected).

No additional configuration is necessary if the deserializers are never powered down during normal operation. If the host microcontroller connected to the serializer never accesses the GMSL1 deserializers, set FWDCCEN = REVCCEN = 0 in registers 0x404 and 0x604.

For all other use case scenarios, one of the following procedures are required to prevent I2C communication failures:

- Use point-to-point I2C communication instead of broadcasting (preferred option).
    - Keep the forward/reverse control channel of each PHY disabled until the host microcontroller is ready to address the individual deserializer. Set FWDCCEN = REVCCEN = 0 in registers 0x404 and 0x604.
    - To allow the microcontroller to communicate with a deserializer:
        - PHY A: Set FWDCCEN = REVCCEN = 1 in register 0x404.
        - PHY B: Set FWDCCEN = REVCCEN = 1 in register 0x604.
    - Disable the bits after I2C communication is complete.
- Lower the I2C slave timeout of the serializer below the deserializers' I2C master timeout.
    - The serializer I2C timeout is set with the G1_SLV_TO register (default: 16ms). Reduce this to be lower than the value of the deserializers' MST_TO timeout setting. Note that this may not be possible with all device combinations (the lowest G1_SLV_TO timeout setting for the MAX96789/91 devices is 1ms).

ADI Confidential for NewVision under NDA
Analog Confidential Distribute Only under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

2.3.5 Serial Link Configuration by Part Number

Not all devices support all the possible link configurations. Link configuration support is shown below (Table 4):

Table 4. GMSL2 Serial Link Configuration by Part Number

Device Family	Part Number	Link Modes Supported							
		GMSL Ports	Single Link PHYA	Single Link PHYB	Single Link PHYC	Single Link PHYD	Splitter Mode	Reverse Splitter Mode	Dual Link Mode
HDMI Serializers	MAX96751	2	X	X			X		X
	MAX96751F	2	X	X			X		X
	MAX96753	2	X	X			X		X
	MAX96753F	2	X	X			X		X
	MAX96751H	2	X	X			X		X
Advanced HDMI Serializers	MAX96763	2	X	X			X		X
	MAX96765	2	X	X			X		X
	MAX96759	2	X	X			X		X
	MAX96761	2	X	X			X		X
	MAX96785	2	X	X			X		X
	MAX96787	2	X	X			X		X
DSI Serializers	MAX96755	2	X	X			X		X
	MAX96755F	2	X	X			X		X
	MAX96755H	1	X						
	MAX96757F	2	X	X			X		X
	MAX96757	2	X	X			X		X
	MAX9283	1	X						
	MAX9285	1	X						
Advanced DSI Serializers	MAX96789	2	X	X			X		X
	MAX96789F	2	X	X			X		X
	MAX96791	2	X	X			X		X
	MAX96791F	2	X	X			X		X
CSI-2 Serializers	MAX9295A	1	X						
	MAX9295B	1	X						
Advanced CSI-2 Serializers	MAX9295D	1	X						
	MAX9295E	2	X	X					X
CSI-2 Camera Deserializers	MAX96784	2	X	X				X	X
	MAX9296A	2	X	X				X	X
	MAX9296B	2	X	X				X	X
CSI-2 Display Deserializers	MAX96756	2	X	X				X	X
	MAX96756F	2	X	X				X	X
	MAX96758F	2	X	X				X	X
	MAX96758	2	X	X				X	X
OLDI Deserializers	MAX96752	2	X	X				X	X
	MAX96752F	2	X	X				X	X

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

	MAX96754	2	X	X				X	X
	MAX96754F	2	X	X				X	X
eDP/DP Deserializers	MAX96772	2	X	X				X	X
	MAX96774	2	X	X				X	X
	MAX96776	2	X	X				X	X
	MAX96778	4	X	X				X	X
CSI-2 Quad Deserializers	MAX96712	4	X	X	X	X			
	MAX96722	4	X	X	X	X			

2.3.5.1 Cables Supported

GMSL2 devices support both 50Ω coaxial and 100Ω differential pair cables (e.g., STP, SPP, STQ, etc.) that meet the GMSL2 Channel Specification. Reference the GMSL2 Hardware Design Guide for more details on supported cables and connecters.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

2.4 Device Power-Up

The GMSL2 device is in power-down mode when either the PWDNB pin is low or any power supply is below its respective power-on reset (POR) threshold. The device begins the power-up sequence after the PWDNB pin is driven high and all power supplies are above their respective POR thresholds. GMSL2 devices do not require power supply sequencing; however, it is recommended to enable all the power supplies followed by the PWDNB pin to precisely control when the device powers on. The Initial Power-Up Sequence applies to all GMSL2 devices; the Link Start-Up Procedure is different for devices operating in GMSL2 Mode and those operating in backward compatible GMSL1 Mode. Follow the applicable procedures as indicated in the steps below.

2.4.1 Initial Power-Up Sequence

The following sequence is performed automatically by the device after the PWDNB pin is driven high and all power supplies are above their respective POR thresholds. Note that there are differences depending on whether the device is in GMSL1 or GMSL2 mode.

- The oscillator is powered up:
  - GMSL1 mode: The internal free-running oscillator is powered up.
  - GMSL2 mode: The crystal oscillator is powered up.
- Internal termination resistance calibration using the 402Ω external resistor connected to the XRES pin is performed.
- Configuration (CFG) Pins levels are sampled:
  - Two-level configuration pins: Pin levels are latched. Internal registers are set according to the latched pin levels.
  - Multi-level CFG pins: CFG pin levels are sampled and latched. Internal registers are set according to the latched pin levels.
- GMSL1/2 Link Start-Up:
  - GMSL1 mode: Continue with the GMSL1 Mode Link Start-Up Procedure.
  - GMSL2 mode: Continue with the GMSL2 Mode Link Start-Up Procedure.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

2.4.2 Link Start-Up Procedure

2.4.2.1 GMSL2 Mode

Following the *Initial Power-Up Sequence*, the local-side control channel (I2C or UART) is functional, and the device registers are writable and readable. The power-up time to this state is given in each device's datasheet.

- GMSL2 devices (i.e., serializer and deserializer) can power up in any order. After power-up, each serial link device sends a beacon signal out to available PHYs to recognize other devices present on the channel.
- A handshake procedure is initiated after successful recognition with the beacon signal.
- Transmitter (Tx) canceller calibration is automatically performed on serializer(s) and deserializer(s) for the forward and reverse channels.
- Equalizer autocalibration is performed on serializer(s) and deserializer(s) to optimize equalizer coefficients and maximize both horizontal and vertical eye openings for the forward and reverse channels. Reference *the Benefits of Adaptive Equalization (AEQ)**Error! Reference source not found.* section for more details.
- Video and control channels are enabled. The LOCK pin and LOCK status register go high on both serializer and deserializer (i.e., link lock is established).

Note: The GMSL2 *Link Start-Up Procedure* will be completed for any channel that meets Channel Specification in a maximum time provided in the device-specific datasheets (refer to the t_{LOCK} parameter). LOCK time may vary due to channel, crystal stability, temperature, and manufacturing tolerance(s).

2.4.2.2 GMSL1 Mode

GMSL2 devices configured for backward compatible GMSL1 mode require separate start-up procedures for serializers and deserializers on the serial link.

- Serializer:
  - The high-speed PLL powers up when a valid pixel clock is detected at the input.
  - Data transmission is initiated after the high-speed PLL locks.
- Deserializer:
  - GMSL1 input detects signal and powers up. The clock and data recovery (CDR) block performs frequency, phase, and word boundary locking.
  - LOCK pin is set high.
  - Control channel access is available from remote side (i.e., over the serial link).

2.4.3 GMSL2 Link Lock

GMSL2 link lock is an automatic bi-directional lock that occurs when a properly connected serializer and deserializer are both powered up (following the procedures in the *Initial Power-Up Sequence* section). In this state, the forward and reverse channels' LOCK pins mirror each other (LOCK pins will go high in all devices). The detection of sync words on the serial link determines link lock.

In GMSL2 mode, the serial link uses the 25MHz crystal or external reference clock (reference the *Clocks and Frequency Reference* section) as the clock source for the serial link. A valid video input (pixel clock) is not necessary to establish GMSL2 link lock.

Maxim Confidential
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Version under NDA

In GMSL1 mode, the input pixel clock is the clock source for the serial link. LOCK is asserted when the deserializer PLLs are locked to the correct serial word boundary alignment. As with native GMSL1 devices, the LOCK pin is available only for the forward channel. Reference GMSL1 device datasheets for further information.

2.4.3.1 LOCK pin

Both serializers and deserializers have an open-drain LOCK output pin. The LOCK pin indicates that the PLLs for the GMSL2 serial link are locked and that the GMSL2 data receive path (i.e., forward channel in serializer, reverse channel in deserializer) has locked to the correct word boundary alignment. Control channels (i.e., I2C/UART, SPI, GPIO, and RGMII) can be used immediately after LOCK is asserted.

Note: The LOCK pin or register MUST be monitored for successful GMSL lock status. Placing a hard-coded wait threshold is NOT recommended. The LOCK status should be obtained prior to continuing with the remaining system programming or actions. If a system level time-out is desired to ensure there isn't a system issue, a time-out can be placed on the system to reset the devices after 1s.

2.4.3.2 Losing Link Lock

Link lock will be lost (i.e., LOCK pins will go low) if the serial link cable is unplugged or there is some other interruption to the system connectivity. In Single Link mode, the link lock will automatically be re-established (following the Link Start-Up Procedure) when the serial link cable is reconnected (i.e., "hotplugged"). Video and control channels will also be automatically re-established.

Note: If the LOCK pin or bit is being monitored for such an interruption, please allow 100ms of recovery time for a clean system relock. Only issue a reset to the link or device if it does not recover after this delay.

Similarly, the link will be lost if either the serializer or deserializer is powered down. For Single Link mode, link lock will automatically be re-established when the device is powered on again, however, video and control channel settings will need to be re-programmed. Follow the appropriate guidance for optimal GMSL link lock time based on the system.

2.4.3.3 Video Lock

Video lock indicates that the deserializer is receiving valid video data from the serializer. After the GMSL2 link has locked, the deserializer video lock sequence will begin. Optionally, the LOCK pin behavior in the deserializer can be changed by a register setting (LOCK_CFG) so that the LOCK pin is asserted only when the deserializer is receiving video (asserted with VIDEO_LOCK).

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

2.4.4 GMSL2 Link Lock Time Optimization

The universal optimization process for all link configurations is to control the timing of the power up or resets of the deserializer (DES) and serializer (SER) devices. Due to timeouts in the link layer, GMSL link lock times (t_{LOCK}) may increase above 160ms or 200ms if DES and SER links are powered up or released from reset with more than 90ms or 70ms delay (t_{RD}) relative to each other. For the timings provided in the link lock time optimization sections, the first value refers to the Group 1 setups and the second to the Group 2 setups. Group 1 contains those systems where the typical lock times in **both** the SER and DES datasheets are less than 55ms. Group 2 are those systems where the typical lock times of **either** the SER or DES are greater than or equal to 55ms.

In certain configurations, devices must be programmed after the power-up (e.g. to set the link rate) and the link must be reset to trigger the configuration change. As the user-issued link resets may interrupt on-going link lock attempts, it is critical to ensure that link resets are issued on both SER and DES sides and are overlapping in time as shown on the diagram below. Without overlapping resets of both SER and DES devices, one of the devices may have to wait to complete its timeout sequence before it becomes available for the next link lock attempt. Overlapping link resets on SER and DES devices should be released within t_{RD} time window relative to each other.

Note: For the timing diagrams in the GMSL2 Link Lock Time Optimization sections, PWDNB indicates the power of the device, whether controlled by PWDNB or VDD. RESET_LINK, not Oneshot Reset, is the preferred method of resetting the link for proper timing control, unless otherwise noted.

This timing method works in all link configurations: single link, splitter, reverse splitter and dual-link.

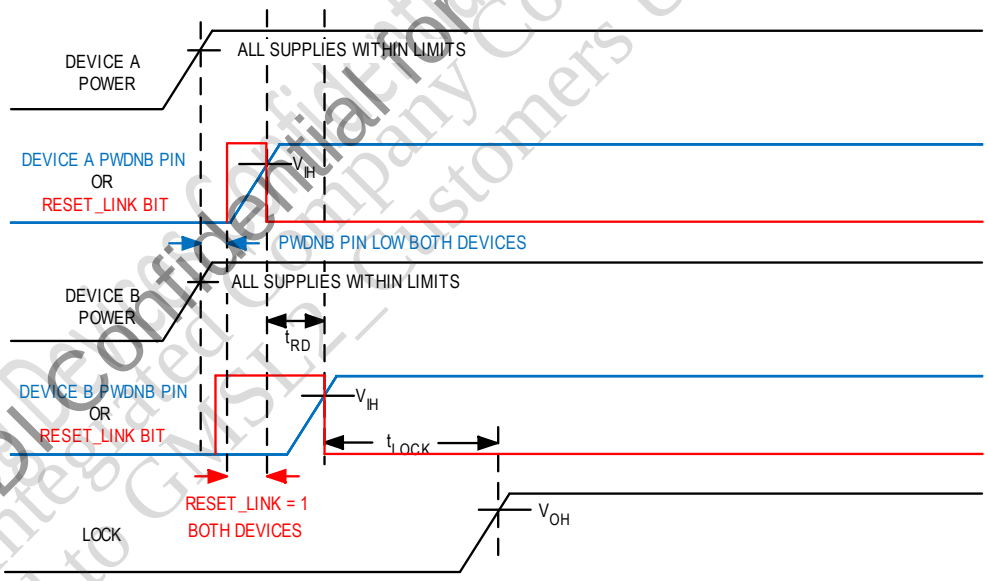


Figure 5. Optimal Initialization for Link Lock Time

ADI Confidential for New Version under NDA
Analog Devices Confidential Distributed to Maxim Integrated Company Confidential Provided to GMSL2_Customers Under NDA

Note: The preferred method for optimal link lock time is to ensure t_{RD} meets the requirement.
The t_{RD} measurement is defined as less than 90ms or 70ms (also noted as 90/70ms), depending on the GMSL device family Group defined above.

The table of expected GMSL link lock behavior is captured in the Link Lock Time Guidance below. But to obtain improved lock times, steps 1 or 2 should be followed to achieve the lock times noted in each. Refer to *GMSL2 Link Configurations* for details on the different link configurations.

1. If t_{RD} is less than the requirement, the datasheet typical lock time will be achieved
2. If t_{RD} timing cannot be controlled or optimized, alternative workarounds for different link configurations are as follows, where exceptions or modifications are noted where applicable:
    a. Single-link configurations
        i. For the MAX96716A/F, MAX96717/F/K, MAX96792A and MAX96793 revision ID = 3 devices, ensure that the SER AUTO_LINK = 1 to achieve the typical lock time. The SER, with AUTO_LINK = 1, should be powered up or released from reset first.
        ii. For the MAX9295A/B and MAX96911D/G, ensure that the DES AUTO_LINK = 1 to achieve the typical lock time. The DES, with AUTO_LINK = 1, should be powered up or released from reset first.
        iii. On all other devices with the AUTO_LINK setting, use the register setting of AUTO_LINK = 1 for both the SER and the DES to achieve the typical lock time.
    b. Splitter configurations
        i. For the MAX96716A/F, MAX96717/F/K, MAX96792A and MAX96793 revision ID = 3 devices, splitter configurations may have increased lock time when t_{RD} exceeds 90/70ms.
        ii. On all other devices with the AUTO_LINK setting, use the register setting of AUTO_LINK = 1 for both DES devices and make sure that both DES devices are powered up or released from reset before the SER for typical lock times. If DES AUTO_LINK bit is set to 0 or SER is powered up/released from reset first, then the lock time may increase when t_{RD} exceeds 90/70ms.
    c. Reverse splitter configurations
        i. For the MAX96717 device family, MAX96793, MAX9295A/B and MAX96911D/G, reverse splitter configurations may have increased lock time when t_{RD} exceeds 90/70ms.
        ii. On all other devices with the AUTO_LINK setting, use the default register setting of AUTO_LINK = 1 for both SER devices and make sure that the SERs are powered up or released from reset before the DES to achieve typical lock times. If SER AUTO_LINK bit is set to 0 or DES is powered up/released from reset first, then the lock time may increase when t_{RD} exceeds 90/70ms.
    d. Dual-link configurations
        i. Dual-link configurations may have increased lock time when the t_{RD} exceeds 90/70ms.
        ii. With this following workaround, the total lock time will be approximately the sum of typical lock times (as if t_{RD}< 90/70ms) for locking the single-link and then relocking in the dual-link mode.
            1. Lock GMSL link in a single link mode first (using AUTO_LINK = 1 setting)
            2. Program both SER and DES to the dual-link mode (by writing LINK_CFG = 0 and AUTO_LINK = 0).
            3. Apply the dual-link settings by executing RESET_ONESHOT on DES and SER. It is critical that the RESET_ONESHOT on the DES and SER are executed within 10ms

ADI Confidential for New Version Under NDA
Analog Devices Confidential Distributed Under NDA
Maxim Integrated Confidential
Provided to Company Confidential, Customers Under NDA

of each other. This can easily be accomplished by having the same ECU write the RESET_ONESHOT to both devices: first write RESET_ONESHOT over the link to the remote device (SER or DES), then immediately write RESET_ONESHOT to the local device (DES or SER).

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

2.4.4.1 Link Lock Time Guidance

The time to lock is influenced by many factors. The link layer is complex, and some devices have an AUTO_LINK or DIS_LOCK_TO configuration bits that can have influence. Therefore, the timing between the SER and DES can be critical for optimum GMSL link lock time.

Note: When there are differing typical lock times on the SER and DES datasheets, use the higher of the two values to determine the expected typical lock time.

The following table provides guidance on how to control the GMSL link lock time using these specific timings and settings.

Link Configuration	GMSL Device Family Group	Other Configuration	t_{RD} (ms)	Sequencing	Typical Maximum (ms)
Any configuration (single link, splitter, reverse splitter, dual-link)	Group 1		<90ms	any	70
Single Link, Glitchless Splitter, Non-Glitchless Splitter, Reverse Splitter			>90ms	Device with (AUTO_LINK=1 AND LINK_CFG = 1/2) -> Device with AUTO_LINK=0/1	70
				SER with (AUTO_LINK=0 AND LINK_CFG = 1/2) OR LINK_CFG = 0/3 -> DES with AUTO_LINK=1/0	160
				DES with (AUTO_LINK=0 AND LINK_CFG = 1/2) OR LINK_CFG = 0/3 -> SER with AUTO_LINK=1/0	>160
Dual link				any	>160
Single Link, Splitter, Reverse Splitter	Group 2		<70ms	any	90
			>70ms	any	>200
		DIS_LOCK_TO = 1		any	200

AUTO_LINK and DIS_LOC_TO configuration bits where applicable
Group 1 are devices that have typical lock time of SER **and** DES < 55ms
Group 2 are devices that have typical lock time of SER **or** DES ≥ 55ms

Table 5. Link Lock Time Guidance

2.4.4.2 POC Link Lock Optimization

POC is common in camera applications where it can be used to synchronize the power up sequence to optimize the lock times. When using POC, the power to the SER is controlled by the DES module. In this case, the SER power must be held off until the DES device is ready to lock to optimize the lock time. This can be after DES power up or after a link reset, whichever is last. The t_{RD} should be less than the allowed t_{RD} requirement. This will optimize the GMSL link lock time to align with the typical lock time. This method could be used in the reverse splitter application.

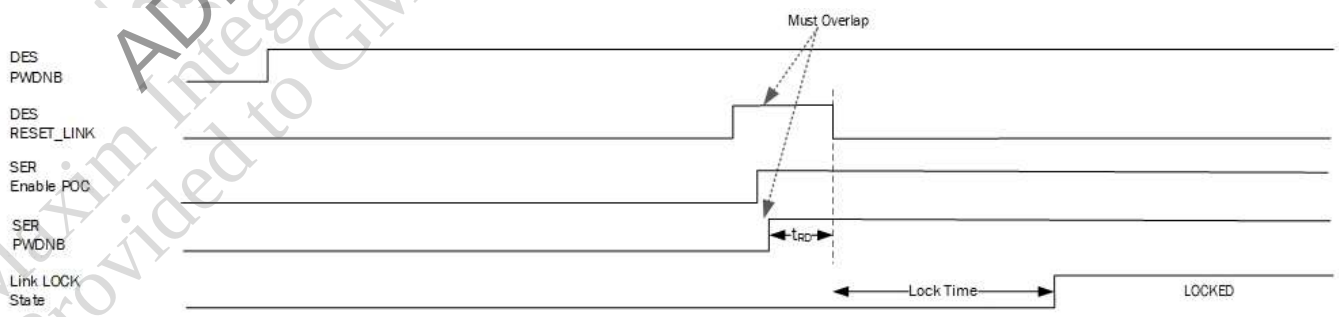


Figure 6. POC Link Lock Optimization

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential
Provided to GMSL2 Company Customers Under NDA

Similar methodology could be applied to the POC display applications where the roles of SER and DES are reversed.

2.4.4.3 Link Reset After Power Up: Link Lock Optimization

In certain scenarios, the device must be configured and the link must be reset to trigger the configuration change. In this case, the system timing can be modified to optimize the relock time by overlapping the RESET_LINK commands of the SER and DES. The RESET_LINK release timings should meet the t_{RD} requirement. If this is controlled by two independent sides, the system can be characterized once combined and the RESET_LINK can be shifted or elongated to ensure the overlap to achieve optimal lock time.

The timing of the initial RESET_LINK can vary based on the individual system after power up, and therefore a couple scenarios are shown below.

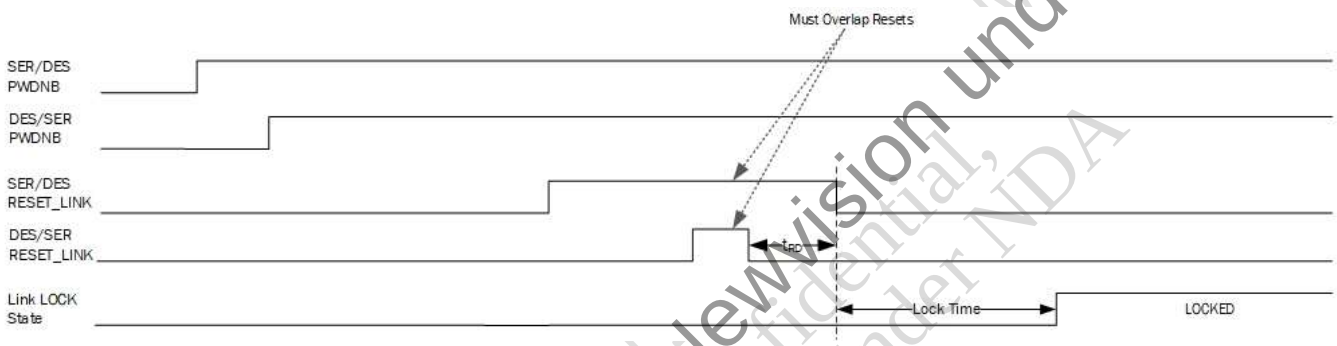


Figure 7. RESET_LINK Timing Optimization

In the case where an initial link lock is achieved before the needed configuration and associated RESET_LINK are applied, the timing might look like Figure 8 where the two RESET_LINK signals will need to overlap and the t_{RD} must still be met to achieve the expected typical lock time.

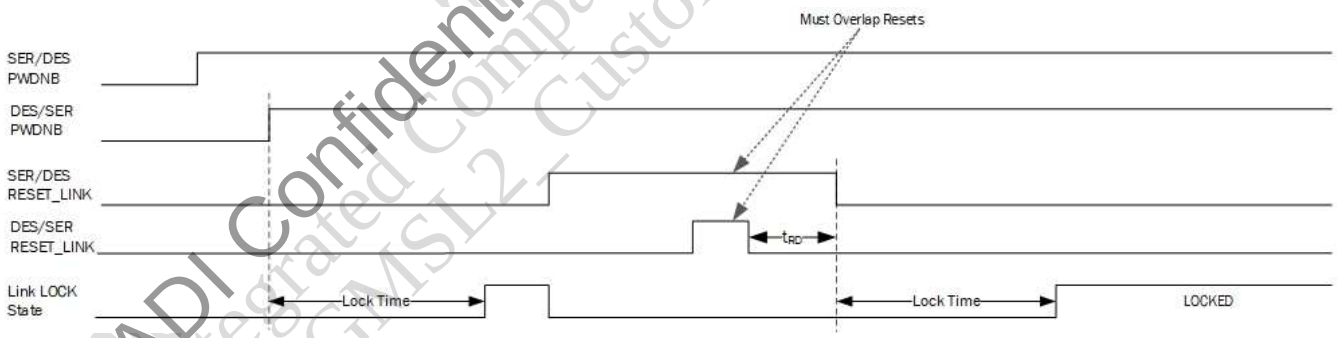


Figure 8. Relock Optimization

ADI Confidential Material
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA
Analog Devices Confidential, Provided under NDA

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

2.4.5 Benefits of Adaptive Equalization (AEQ)

Adaptive Equalization addresses critical aspects of serial link implementation and ensures highest possible link performance. Manual equalization is not recommended since it fails to dynamically respond to factors that affect channel quality.

1. AEQ automatically adapts to changes in channel characteristics. Optimal equalization requires dynamic response to variations in IC process(es), package(s), temperature, cable insertion and return loss, PCB layout, and PoC performance.
2. Equalization is non-linear; nonoptimized equalizer settings will have a non-linear effect on eye opening and link margin. AEQ corrects for link margin to ensure optimum link quality.
3. Changes to equalization alters the signal-to-noise ratio on the bi-directional GMSL2 serial link non-linearly across the frequency band. Fixed equalization will result in a loss of correlation between link margin and channel quality.
4. Manual programming of the canceller registers is inexact. If the canceller is not optimized, signal may be misinterpreted as a noise source. The reduction of the signal-to-noise-ratio would negatively impact link performance and link BER.

2.4.6 Periodic Adaptation

The GMSL2 PHY for both forward and reverse channel periodically re-adapt the equalization to optimize receive-side gain and equalization settings for different environmental and use-case conditions. This allows the equalizer to compensate for changes in the channel due to temperature, aging cables, PoC loads, cable bending, and/or other external factors.

Channel characteristics are dynamic and environmentally dependent. The periodic adaptation (which default runs at a 1Hz rate) allows for the equalizer to continually track and adapt to these channel changes. This is illustrated in the eye diagrams (Figure 9 and Figure 10) below.

Analog Devices Confidential Distribute Outside NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Version under NDA

2.4.6.1 Periodic Adaptation Performance Impact

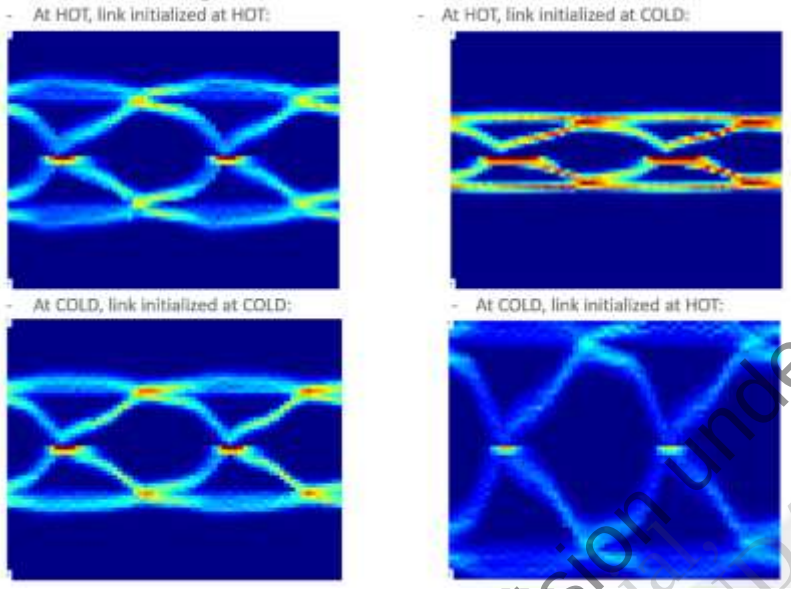


Figure 9. Periodic Adapt not Enabled

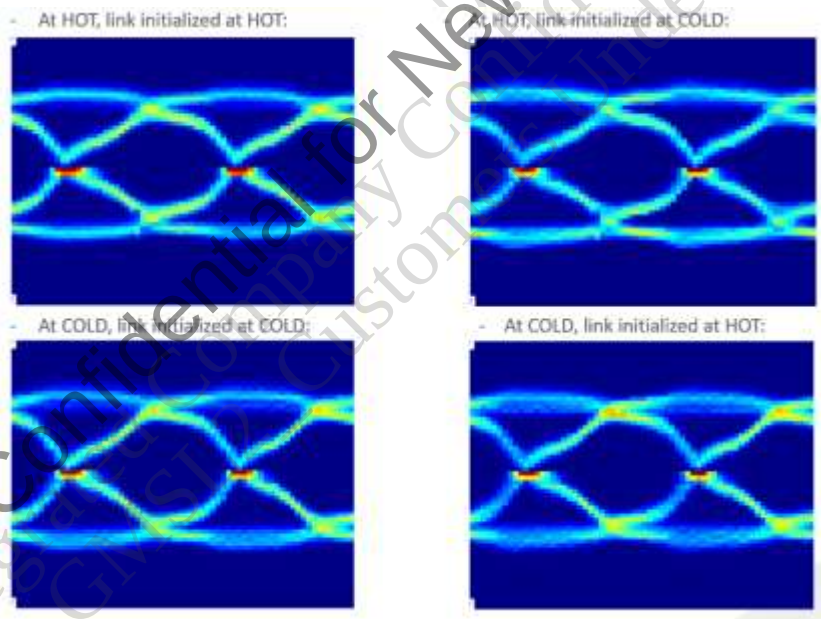


Figure 10. Periodic Adapt Enabled

With periodic adaptation enabled, the eye is continually optimized as the temperature changes. Without periodic adaptation enabled, variations in cable temperature from the initial link condition affect the eye-opening. A non-optimized eye-opening leads to lower link BER.

ADI Confidential for Newvision under NDA
Analog Devices Maxim Integrated Company Confidential Customer Provided to GMSL2 Customers under NDA

2.5 Resets

Registers that affect GMSL2 PHY operation should be programmed when the link is held in reset.

A link reset must be used after programming:

- TX_RATE
- RX_RATE
- CXTP_A/B
- AUTO_LINK
- LINK_CFG
- GMSL1/GMSL2 mode

Note: There are other situations when a link reset may be needed which are described in their relevant sections.

2.5.1 Types of Resets

There are three different reset bits available for GMSL2 devices:

2.5.1.1 Reset All

Reset All is a full reset that resets the link, all registers, and all digital and analog blocks. Writing RESET_ALL = 1 (cleared when written) or toggling the PWDNB pin low then high will perform a Reset All.

2.5.1.2 Oneshot Reset

The Oneshot Reset is a self-clearing bit that resets the link without resetting registers. This can only be performed by writing RESET_ONESHOT = 1. Use this reset after making a change that affects the GMSL2 serial link, such as changing the link rate.

2.5.1.3 Reset Link

Reset Link is similar to Oneshot Reset, but it is not self-clearing. This bit should be used if the link should be held in a reset state to suspend connect attempts to the remote side. After this bit is set, all local registers are still available, but the remote device is not accessible. This bit should not be set on the remote device: setting Reset Link on the remote device will prevent write access on the link and the reset will be unable to be cleared. Enable by writing RESET_LINK = 1; release the reset by writing RESET_LINK = 0.

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribution Only under NDA
Maxim Integrated Company Confidential, Customers Under NDA
Provided to GMSL2 Customers

2.6 GMSL2 Link Protocol

GMSL2 is a programmable fixed rate packet-based protocol, flexibly sharing link bandwidth between video data and multiple bi-directional communications channels. Dynamic bandwidth allocation allows active channels to share full link bandwidth; inactive channels do not consume any link bandwidth. Additionally, maximum packet size is limited to prevent a single channel from over-utilizing link bandwidth.

2.6.1.1 Encoding

The GMSL2 serial link uses a proprietary 9b10b encoding method. This encoding method features:

- Less encoding overhead (11%) compared to traditional 8b10b encoding (25%).
- DC-balance: The number of 1s and 0s in any given window is bounded to +/- 9.
- A maximum run length (consecutive number of 0s or 1s) limited to 7.
- Eight special symbols used for link synchronization, framing (packet start/stop indication), and control purposes. The Hamming distance between special symbols is at least 3, so these are robust to errors (i.e., the special symbols will not accidentally transform into another special symbol with 1- or 2-bit errors in the 10-bit symbol).
- Special sync words for word boundary locking contain an exclusive comma sequence that cannot exist in regular random encoded data stream starting from any bit position.

2.6.1.2 Scrambling

The GMSL2 serial link features synchronous scrambling to improve clock and data recovery robustness. This also improves EMI performance by reducing any pattern-dependent EMI emission. The scrambler in the transmitter and the descrambler in the receiver are synchronized with sync words.

2.6.1.3 Special Symbols

Special symbols (individually and in combination) are used as packet delimiters and headers. These are selected to keep GMSL2 link robust (i.e., tolerant to bit errors) as much as possible. All special 9b10b symbols have 3-bit Hamming distance to each other; it is impossible for a special symbol to transform into another special symbol with 1- or 2-bit errors.

2.6.1.4 Total Link Bandwidth

Total link bandwidth used by all different communication channels cannot exceed the fixed available link bandwidth. In typical use cases, available link bandwidth exceeds the bandwidth requirement. Idle packets are used to fill unused link bandwidth.

2.6.2 Packet Protocol

GMSL2 serial links use a packet-based protocol comprising various packet types to transmit information across the link. The packet types are summarized below.

Note: The GMSL2 forward and reverse channels use the same protocol.

Maxim Integrated Confidential, Distribution Only Under NDA
Provided to GMSL2_Customers Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
Maxim Integrated Confidential, Distribution Only Under NDA
Maxim Integrated Confidential, Distribution Only Under NDA
Maxim Integrated Confidential, Distribution Only

2.6.2.1 Video Packets

Video packets are used to transmit up to four concurrent video streams over the GMSL2 serial link. Each video packet consists of 36 pixels and a 1-bit sequence number that allows the receiver to detect dropped packets.

During video blanking time, pixel data values are not transmitted to save bandwidth. By default, HS, VS, and DE (HVD) values are transmitted from the serializer so that the receiver can exactly reconstruct the same blanking time that the serializer receives on the video input. This is needed when driving a deserializer with OLDI output.

Optionally, the serializer can be programmed to "remove the heartbeat" (LIM_HEART = 1). In this mode, the serializer does not transmit HVD packets to minimize bandwidth usage. This mode can be used when driving a deserializer with CSI-2 or eDP/DP output, as those deserializers do not require HVD values. In this case, SEQ_MISS_EN = 0 and DIS_PKT_DET = 1 should be programmed in the deserializer so that it does not expect HVD packets during video blanking.

By default, video packets do not have embedded CRC. When bandwidth is available, this can be enabled to attach a 16-bit CRC value to each video packet. This is checked by deserializer device and reported.

2.6.2.2 Control Channel Packets

Control packets are used for communications channels that use less than 100Mbps of bandwidth. This includes I2C, UART, SPI, GPIO, Audio, and internal Info Frame packets. Info Frame packets are internal control packets used to transmit information between serializer and deserializer (e.g., video PCLK PLL settings, audio PLL settings, etc.).

Control packet types are differentiated by header formats. Each control packet is tagged with a 4-bit sequence number and includes a 16-bit CRC value by default. These are used for error detection and correction. See the CRC Error Detection and ARQ Error Correction section for additional information.

2.6.2.3 Sync Words

Sync words are periodically transmitted packets used for word boundary locking, lock validation, and scrambler synchronization. Sync words are also used to align two GMSL2 lanes when the serial link is configured in dual link mode.

2.6.2.4 RGMII/RMII Packets

RGMII/RMII packets are used to tunnel RGMII and RMII data. CRC is disabled by default for this packet type, as Ethernet has error-detection capability at higher protocol layer. However, CRC can optionally be enabled if bandwidth is available.

2.6.2.5 Idle packets

Idle packets are transmitted to fill the link bandwidth when there is no data to transmit. Both the header and payload data of idle packets are scrambled to minimize EMI. Idle packets are decoded and checked for errors, which are reported to the IDLE_ERR_FLAG register. See the GMSL Idle Packet Errors section for additional information.

ADI Confidential for Review under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2 Customers under NDA

2.6.3 Priority-Based Packet Scheduler

GMSL2 devices support multiple communication channels in addition to the forward channel video. Available bandwidth is flexibly shared between the various channels with dynamic bandwidth allocation.

GMSL2 devices use a bandwidth allocation scheduler that acts as an arbiter if multiple channels (i.e., video or control channels) attempt to simultaneously send packets over the link. The scheduler uses a channel-based priority setting and a calculated running average of the bandwidth usage of each type of communication channel to schedule transmissions over the link and throttle channels if needed.

Link bandwidth sharing is based on pre-defined share ratios and the recent bandwidth usage of each channel. GMSL2 devices track averages of recent bandwidth usage for each channel compared to assigned bandwidth share ratios. When deciding which packet to transmit, the scheduler will select the channel with lowest usage ratio from the pending requests.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

3 Spread-Spectrum Clocking (SSC)

Spread Spectrum Clocking (SSC) provides enhanced mitigation of electromagnetic interference (EMI) emitted from devices and interconnections. GMSL2 serial links offer exceptional EMI performance; however, it is recommended to use SSC when possible to reduce emission peaks for additional margin. This option is a feature of all GMSL2 devices and is also available in GMSL1 mode.

Spread spectrum is available for the following interfaces:

- Forward serial link (GMSL1 and GMSL2 mode)
- Reverse serial link
- Serializer camera clock reference output
- Deserializer OLDI outputs
- Deserializer eDP/DP outputs

Configuration is similar for each of the above interfaces. Programming via the GMSL2 registers is described in the following sections.

3.1 SSC Operation

The SSC uses a constant frequency 6GHz Clock Multiplier Unit (CMU) clock and a phase interpolator. A digital block generates the phase commands to modulate the transmitter clock. Linear frequency modulation requires quadratic phase command generation. The digital logic supports five levels of SSC generation all at 25KHz. An example center frequency spread is illustrated below in Figure 11.

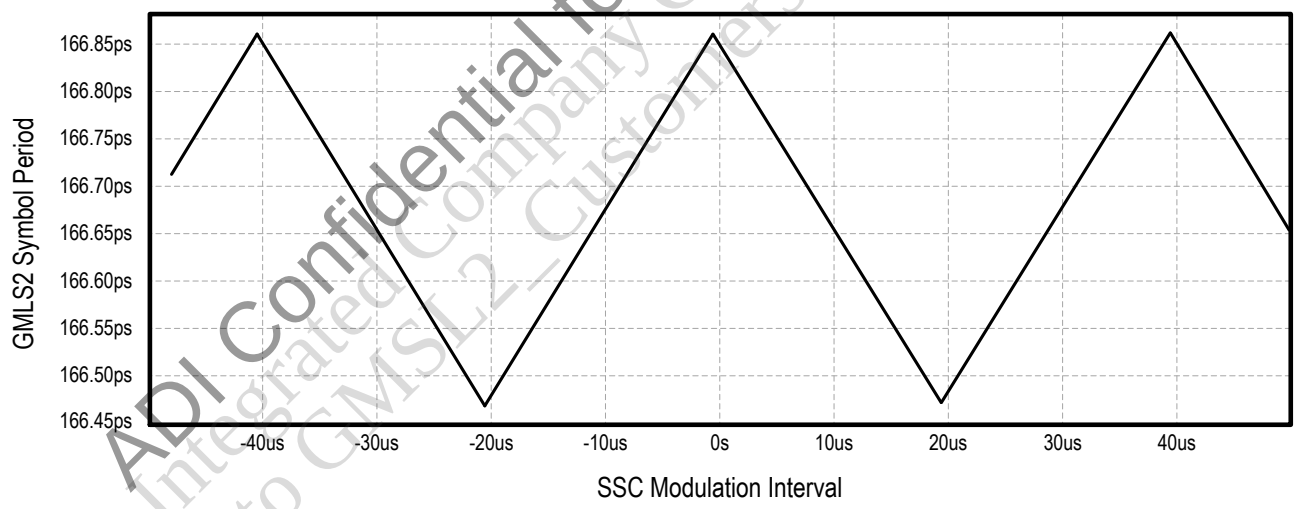


Figure 11. Example SSC Center Frequency Spread

ADI Confidential for review under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
Do not distribute or duplicate

3.2 SSC Configuration

Enabling spread spectrum for a GMSL2 serial link requires configuring registers controlling the phase command generation and maintenance of the 25kHz triangle waveform. The forward and reverse channel spread spectrum generator must be disabled before configuring its settings and then re-enabled once the settings are configured.

SSC can be individually enabled and configured for various interfaces, allowing for user flexibility in their target application. Refer to the register settings in the tables below to configure the spread spectrum transmitter on either side of the link. For GMSL2 devices, writing these values to the serializer registers will implement spread spectrum on the forward serial channel, while writing these values to the deserializer registers will implement spread spectrum on the reverse serial channel.

The maximum spread for the GMSL2 serial link is 2530 ppm.

3.2.1 Forward and Reverse Channel SSC

Care must be taken to prevent large phase transients on the transmit clock. The procedure to enable SSC on the GMSL2 forward/reverse serial link is as follows:

1. Disable SSC generator: RLMS71[0] = 0.
2. Configure SSC generator per the table below
3. Enable SSC generator: RLMS71[0] = 1.

Table 6. GMSL2 Tx SSC Configuration for Fixed PPM

SSC PPM	SSC F_c (kHz)	RLMS 64	RLMS 70	RLMS 71	RLMS 72	RLMS 73	RLMS 74	RLMS 75	RLMS 76	RLMS 77
268	25	0x03	0x07	0x02	0xC9	0x02	0xF9	0x01	0x00	0x00
580	25	0x03	0x06	0x02	0xAB	0x00	0x63	0x07	0x00	0x00
970	25	0x03	0x03	0x02	0xAB	0x00	0x63	0x07	0x00	0x00
1750	25	0x03	0x01	0x02	0xF9	0x00	0x2C	0x05	0x00	0x00
2530	25	0x03	0x01	0x02	0xAB	0x00	0x63	0x07	0x00	0x00

After configuration is complete, enable the generator by writing “1” to RLMS71[0].

Note: Disabling the GMSL2 serial link spread spectrum may cause loss-of-lock. It is recommended to perform a link reset if link spread spectrum is disabled. See the *Resets* section for additional information.

3.2.2 Output Interface SSC

To enable spread spectrum for other available interfaces, the config_spread_bit_ratio[2:0] bits of register DPLL_3 must be configured, and the config_force_enable_ss bit of register DPLL_3 must be set. Once the configuration and enable bits are set, the new spread spectrum settings will take effect. Note that the register addresses may be different for each interface. See Table 7 below for register configuration.

Table 7. Serializer/Deserializer Output Spread Spectrum Register Configuration

Output Interface	Register Name (Serializer/Deserializer)	Register Address	config_spread_bit_ratio[2:0] bit value (spread percent)
eDP/DP	EDP_DPLL_3 (Deserializer)	0x6003	000····· 0% (off)
OLDI	OLDI_DPLL_3 (Deserializer)	0x0D03	001····· 0.25%

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Provided to Customers Under NDA

Reference Clock	REF_DPLL_3 (Serializer)	0x0A03 | 0x1A03	010····· 0.5% (default) 011····· 1% 100····· 2% 101····· 4% 110····· 4% 111····· 4%
	REF_DPLL_3 (Deserializer)	0x0A03 | 0x1A03	

Note: GMSL2 spread spectrum does not propagate downstream; each stage (i.e., forward/reverse link and serializer/deserializer outputs) must be enabled separately.

Note: For eDP/DP output spread spectrum, 0.25% spread (config_spread_bit_ratio[2:0] = 0b001) is the only allowable setting. Any other settings (greater than 0.25%) will be in violation of the DP spec, which may result in malfunctioning of some DP sink devices.

3.2.2.1 Video Output Interruption

If the video output is interrupted after the deserializer spread spectrum is programmed, the output spread spectrum will reset to default or off, but the register value will not reflect this behavior. The DPLL_3 register must be re-written via I2C/UART to the desired spread spectrum value.

3.2.3 GMSL1 Spread Spectrum

GMSL2 devices operating in GMSL1 mode can enable SSC. In GMSL1 mode, the spread spectrum percentage is limited by the serial link bit rate (Table 8).

Spread spectrum can be enabled in GMSL2 devices operating in GMSL1 mode by writing to the SSEN bit in the GMSL1 register block. By default, spread spectrum is not enabled at power-up (SSEN = 0). Writing SSEN = 1 enables spread spectrum and establishes a default 0.5% spread, which is compatible with all serial link bit rates (see Table 8). Spread spectrum can alternatively be enabled with the config_force_enable_ss bit. To adjust the percentage of the spread in GMSL1 mode, adjust the config_spread_bit_ratio bit of register DPLL_3 accordingly.

Table 8. GMSL1 Spread Percentage Limitation

Bit Rate	Max Spread
< 1Gbps	4% (40k ppm)
1Gbps < bit rate < 2Gbps	2% (20k ppm), register max 100
> 2Gbps	1% (10k ppm), register max 011

The video must be received by the serializer before modifying the GMSL1 link spread percentage. If the video is absent or interrupted (PCLKDET = 0), the DPLL_3 registers will be reset. The reset of the DPLL_3 registers will result in different behavior depending on what bits were used to enable SSC:

1. If config_force_enable_ss is used to enable spread, spread will be disabled.
2. If SSEN bit is used to enable spread, any adjustments made with the config_spread_bit_ratio bit will be reset and spread will default to 0.5%.

It is recommended to use SSEN = 1 if 0.5% spread is sufficient for the system. If spread must be a higher value, then use config_force_enable_ss along with config_spread_bit_ratio in the DPLL_3 block (see Table 9).

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential, Distribution Only
Provided for GMSL2 Customers Under NDA

Table 9. GMSL1 Link Spread Spectrum Registers

Output Interface	Register Name (Backward Compatible Serializers)	Register Address	config_spread_bit_ratio[2:0] bit value (spread percent)
GMSL1 Link Spread	AUD_DPLL_3 (HDMI with Compression)	0x0B03	000····· 0% (off) 001····· 0.25% 010····· 0.5% (default) 011····· 1% 100····· 2% 101····· 4% 110····· 4% 111····· 4%
	AUD_DPLL_3 (DSI Serializer) PHY A	0x1B03	
	AUD_DPLL_3 (DSI Serializer) PHY B	0x1C03	

Note: System lock may be lost if spread spectrum is enabled or disabled. After a 5ms wait time, link will relock automatically if video is received.

3.2.3.1 GMSL1 Devices

Spread spectrum for GMSL1 devices should only be enabled on the serializer; the spread will propagate from the serializer through the link to the deserializer outputs. Do not enable on the deserializer unless it is a GMSL2 deserializer in GMSL1 mode.

3.2.3.2 GMSL2 Devices in GMSL1 Mode

If the serializer and deserializer are both GMSL2 devices in GMSL1 mode, spread does not propagate through the link. Spread spectrum must be enabled separately as described in the SSC Configuration section above.

ADI Confidential for review under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

4 GMSL1 Legacy Mode (Backward Compatibility)

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

5 Power Manager and Sleep Mode

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

6 Clocks and Frequency Reference

6.1 Overview

GMSL2 devices require a precise, low-jitter external frequency reference. This frequency reference is multiplied and used to establish the GMSL link clock, other internal clocks, and, in some cases, clocks associated with external video or peripheral interfaces.

Integrated oscillators are included in all GMSL2 devices. Most frequency reference applications require only the addition of an external crystal and associated passive components. Alternatively, the on-chip oscillator can be overdriven by an external oscillator or clock source. This enables multiple devices to share a common frequency reference and potentially reduces the number of crystals required in a complex system.

Select GMSL2 serializers can share the crystal/external clock reference by utilizing a reference clock output (RCLKOUT). This arrangement eliminates the need for an additional external crystal or oscillator for a second serializer. RCLKOUT can be 25MHz (same as the reference clock input) or divided by two or four.

For frequency outputs other than 25MHz (or the divided by two and four options), select GMSL2 serializers can utilize a DPLL to generate and output a user-configured clock output frequency (DPLL_OUT). The DPLL uses the 25MHz reference clock as an input and generates either a predefined or arbitrary frequency based on register programming. This frequency is output at the corresponding GPIO pin.

GMSL2 devices can output either RCLKOUT or DPLL_OUT at a time; these signals are output from the same device pin.

Refer to Feature Availability by Device Family for RCLKOUT and DPLL availability and pinout by part number; refer to device-specific datasheets for specifications and additional information.

6.2 Architecture

The frequency reference for GMSL2 devices is typically realized using an external 25MHz crystal in conjunction with the on-chip oscillator to create an accurate, low-jitter reference. This reference is distributed throughout the GMSL2 device to establish the link and generate other required clocks. An external 25MHz clock source, such as another GMSL2 device, can alternatively be used in the place of the crystal. When using an external oscillator or clock source, the frequency precision and jitter must be within the bounds specified in the electrical characteristics of the device.

A simplified diagram of the frequency reference system is shown in Figure 12. The diagram shows both on-chip and external circuit elements.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distributed to Maxim Integrated Company Confidential Provided to GMSL2 Customers Under NDA

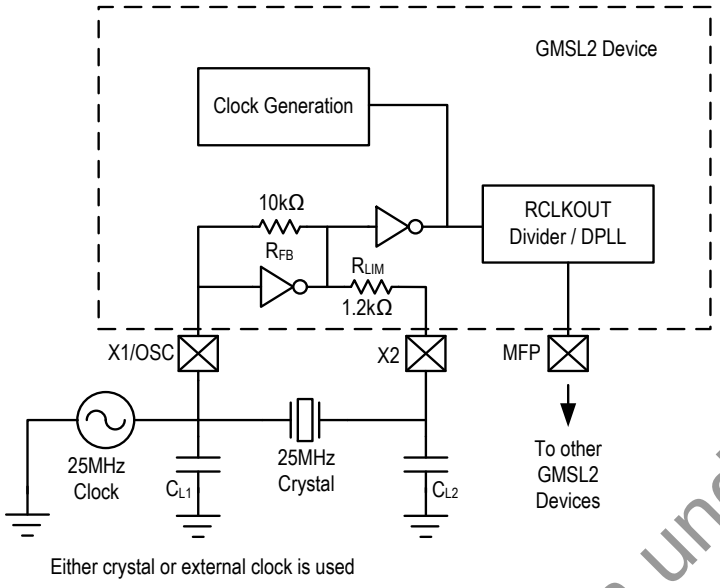


Figure 12. Simplified Diagram of Frequency Reference System

In addition to driving internally generated clocks, the reference can be driven off-chip through either a programmable integer divider or as the reference to a programmable PLL. Here, a single crystal can provide the frequency reference for multiple GMSL2 devices and/or other parts. The RCLKOUT/DPLL block diagram is shown below in Figure 13. Note that the RCLKOUT and DPLL_OUT frequencies output to the same pin, which varies by device. Refer to Feature Availability by Device Family and device-specific datasheets for pinout information.

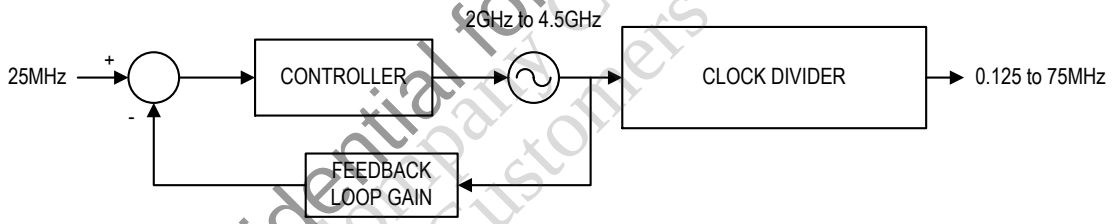


Figure 13. RCLKOUT/DPLL Architecture

The RCLKOUT Configuration section below provides additional details regarding the use and configuration of the RCLKOUT and DPLL functions.

ADI Confidential for new version under NDA
Analog Devices Confidential, Untitled
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

6.3 Operation

Depending on the architecture of a given system, either a crystal or external precision clock source must be used to provide a frequency reference to the GMSL2 device. The following information details some of the key considerations involved in successfully implementing either a crystal- or external clock-based frequency reference. Note that no device programming is required to accommodate one operating mode versus the other.

6.3.1 Crystal Mode

When deriving the frequency reference using a crystal, a device with the required frequency and tolerance (i.e., 25MHz ±200ppm) as well as environment characteristics must be selected to ensure compatibility and reliable performance.

The implementation of the crystal circuit is illustrated in Figure 12. The values of C_{L1} and C_{L2} must be selected based on the specified load capacitance of the crystal and the estimated parasitic capacitances of the PCB and GMSL2 device. An example of the typical parasitic capacitances of the GMSL2 X1/OSC and X2 pins is given in Table 10. Refer to device-specific datasheets for the capacitance of these pins, as they vary by device. Note that the feedback resistor and limit resistor are included on-chip. The selected crystal is connected between the X1/OSC and X2 pins.

Calculation of the load capacitors is discussed in detail in the Configuration and Calculations section below.

Table 10. Example DC Characteristics of X1/OSC and X2 Pins

REFERENCE CLOCK INPUT REQUIREMENTS (CRYSTAL) (X1/OSC, X2)		
PARAMETER	**SYMBOL**	**VALUE (TYP)**
X1/OSC Input Capacitance	C_{IN_X1}	3pF
X2 Input Capacitance	C_{IN_X2}	1pF
Internal X2 Limit Resistor	R_{LIM}	1.2kΩ
Internal Feedback Resistor	R_{FB}	10kΩ
Transconductance	g_M	28mA/V

Proper layout of the crystal circuitry is important to achieving the best possible performance. *Figure 14* provides an example layout of the crystal and associated components. The crystal and associated external load capacitors should be placed near the X1/OSC and X2 pins of the GMSL2 device to reduce interconnect capacitance and susceptibility to noise coupling. If preferred, the crystal can be placed on the opposite side of the board from the GMSL2 device. There should be a solid ground plane in the region of both the crystal and GMSL2 device, and the crystal's ground terminals should be connected to the ground plane with minimal inductance. Finally, noisy components and interconnects should not be arranged near the crystal or associated interconnects. Length matching and impedance of the X1/OSC and X2 interconnects is not critical.

ADI Confidential for NDA Version under NDA
Analog Confidential distribution
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

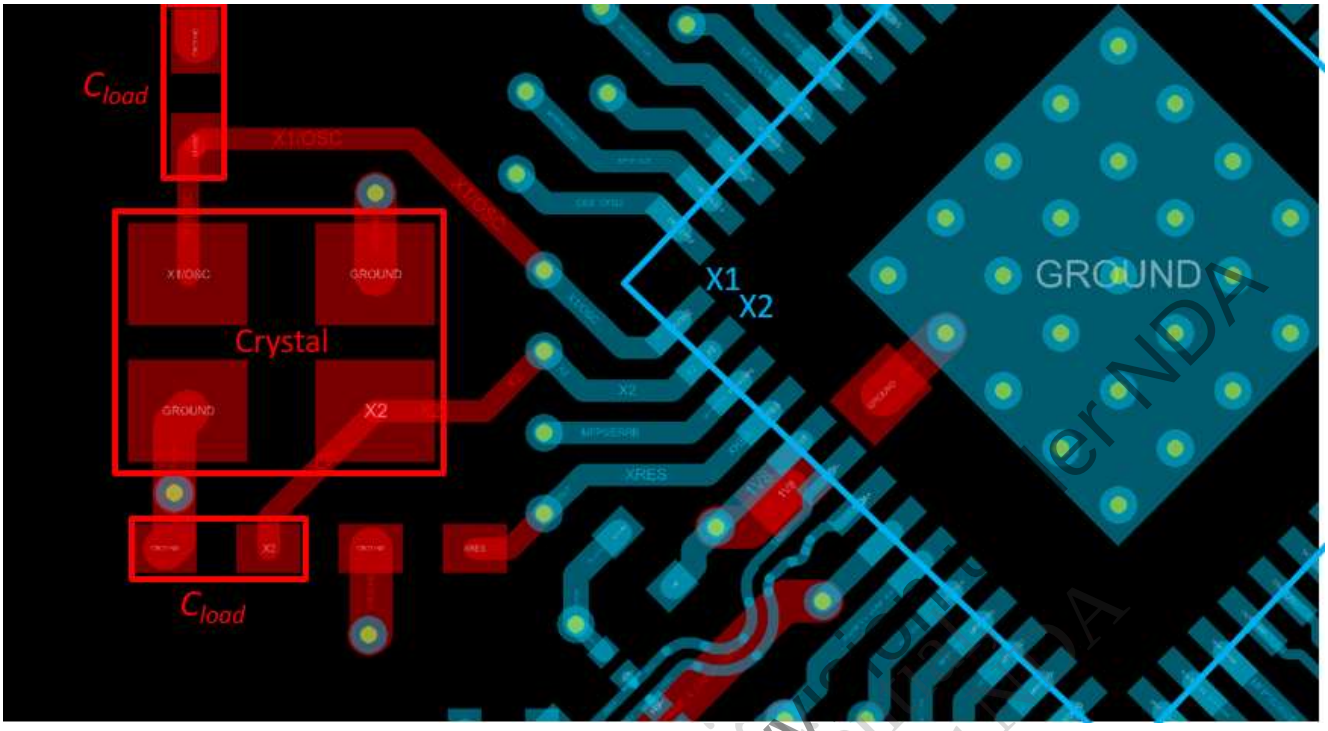


Figure 14. Example Crystal Layout

6.3.2 External Reference Mode

The external load capacitors and crystal can be omitted if the frequency reference is supplied by an external clock source such as a TCXO or the RCLKOUT function of a GMSL2 device. The external clock source must be compliant with the specifications detailed in Table 11 and Table 12. Note that the maximum signal level is constrained by V_{DDIO}. Assuming that the clock source uses a CMOS driver and is powered by the same V_{DDIO} as the device that is being driven, the clock signal can be DC coupled to the X1/OSC pin of the GMSL2 device. For example, a GMSL2 device's RCLKOUT function can be directly connected to another GMSL2 device's X1/OSC pin assuming that the two devices use a common V_{DDIO}. The required frequency accuracy and tolerance are identical to the crystal specifications (i.e., 25MHz ±200ppm). Note that the jitter, duty cycle, and fall time must all be considered when using an external clock source.

Table 11. External Clock DC Characteristics

REFERENCE CLOCK REQUIREMENTS (EXTERNAL INPUT ON X1/OSC, X2 UNCONNECTED)			
PARAMETER	**MIN**	**TYP**	**MAX**
High-Level Input Voltage	0.9V		V_{DDIO}
Low-Level Input Voltage			0.4V
Input Impedance		10kΩ	
X1/OSC Input Capacitance		3pF	

Maxim Integrated Confidential
Provided to GMSL2_Customer Under NDA
ADI Confidential Under NDA

Table 12. External Clock AC Characteristics

REFERENCE CLOCK REQUIREMENTS (EXTERNAL CLOCK INPUT ON X1/OSC, X2 UNCONNECTED)			
PARAMETER	**MIN**	**TYP**	**MAX**
Frequency		25MHz	
Frequency Stability + Frequency Tolerance			±200ppm
Input Jitter		600ps (p-p)	
Input Duty Cycle	40%		60%
Input Fall Time			4ns

Table 13 includes an example of the typical RCLKOUT characteristics of a GMSL2 device (MAX9295D). The jitter and RCLKOUT fall time are within the specified bounds, confirming that this device's RCLKOUT meets the requirements to be used as the frequency reference source for another GMSL2 device. Note that in the case of GMSL2 devices, the drive strength of the RCLKOUT MFP pin must be specified appropriately to achieve the desired fall time depending on V_{DDIO}. A similar set of specifications should be reviewed while evaluating any external clock source.

Table 13. Example RCLKOUT Characteristics (MAX9295D)

REFERENCE CLOCK REQUIREMENTS (EXTERNAL CLOCK INPUT ON X1/OSC, X2 UNCONNECTED)				
PARAMETER	**CONDITIONS**	**MIN**	**TYP**	**MAX**
Frequency	Crystal or reference clock input		25MHz	
Rise Time	20% to 80%, C_L = 10pF			4ns
Fall Time	80% to 20%, C_L = 10pF			4ns
Jitter	C_L = 10pF, rising or falling edge f_{REFOUT} = 12.5MHz		210ps (p-p)	

In contrast to crystal mode, the frequency reference is not guaranteed to automatically be available at power-up in external reference mode. For example, if the RCLKOUT function of another GMSL2 device is used to provide the frequency reference, that device's RCLKOUT function must be enabled prior to the frequency reference being operational.

The layout of a design that uses an external frequency reference must be carefully considered. The clock should be routed so that it is not susceptible to picking up noise or corrupting other noise-sensitive functions. The interconnect should be as short as possible, and a solid ground plane should be used in the vicinity of the clock source and along the length of the interconnect. A provision should be included for a source termination resistor.

6.3.3 Frequency Reference Debugging

Proper operation of the frequency reference is required for GMSL2 link functionality. If a system is not achieving the anticipated operation and power supplies have been checked, verify that the frequency reference is operational.

Test the frequency reference by probing the X2 pin with a low capacitance, high impedance probe. If the frequency reference is functioning correctly, X2 should toggle at 25MHz. If not, inspect the associated solder joints, clean the PCB, and consider replacing the crystal and load capacitors. Finally, confirm that the ESR of the crystal is within the bounds supported by the GMSL2 device and verify that the load capacitance values employed are consistent with the requirements of the crystal.

Analog Devices Confidential Division only
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2 Customers Under NDA
Confidential for Newvision under NDA

6.4 Configuration and Calculations

No programming is required to use the frequency reference in either crystal mode or external clock source mode. At power-up, the frequency reference circuitry is enabled by default; its output is then routed to enable clocking of all sub-systems that are enabled by default.

The RCLKOUT function is not enabled by default; to use this function, follow the applicable configuration process detailed in the RCLKOUT Configuration section.

6.4.1 Evaluating Crystal Specific Components and Parameters

The crystal must adhere to specified bounds to ensure reliable and accurate frequency reference operation. This includes the absolute frequency (i.e., 25MHz for GMSL2 devices), the frequency tolerance at room temperature, and the frequency drift over the anticipated operating temperature range of the crystal. Additionally, the crystal should have appropriate reliability qualifications.

When implementing the crystal circuit, the total load capacitance (as specified in the device datasheet) must be considered. The selected external load capacitors should account for the parasitic capacitance in the system to guarantee that the most accurate absolute frequency precision is attained.

ESR and crystal power dissipation must also be considered to ensure that the oscillator provides reliable operation across the expected environmental conditions.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Limited
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

6.4.1.1 Calculating Load Capacitor Values

The following example summarizes the process of selecting the required load capacitance for optimal frequency precision. First, confirm the required load capacitance (C_L) from the crystal datasheet. Then, determine the external load capacitor values (C_{L1} and C_{L2}) to achieve the proper total C_L as specified in the crystal datasheet. This process is detailed below.

$$C_{X1_total} = C_{IN_X1} + C_{L1} + C_{X1_trace}$$

$$C_{X2_total} = C_{IN_X2} + C_{L2} + C_{X2_trace}$$

$$C_L = \frac{C_{X1_total} * C_{X2_total}}{C_{X1_total} + C_{X2_total}}$$

Where:

C_{IN_X1} = X1/OSC input capacitance from GMSL device datasheet
C_{X1_trace} = X1/OSC trace capacitance from layout
C_{L1} = X1/OSC external load capacitor
C_{IN_X2} = X2 input capacitance from GMSL device datasheet
C_{X2_trace} = X2 trace capacitance from layout
C_{L2} = X2 external load capacitor
C_L = total load capacitance specified in crystal data sheet

6.4.1.1.1 Example Load Capacitor Calculation

Assume the following:

C_L = 18pF (from crystal data sheet)
C_{IN_X1} = 3pF (from GMSL data sheet)
C_{X1_trace} = 2pF (estimated PCB capacitance)
C_{IN_X2} = 1pF (from GMSL data sheet)
C_{X2_trace} = 2pF (estimated PCB capacitance)

The external load capacitor value will be approximately twice the specified load capacitance of 18pF. A commonly available capacitor value that is consistent with this requirement is 33pF. Assuming that C_{L1} = C_{L2} = 33pF, the total C_L can be calculated to be around 18.5pF, which is in good agreement with the requirements.

Assume C_{L1} = C_{L2} = 33pF and calculate C_L

C_{X1_total} = 38pF
C_{X2_total} = 36pF
C_L ≈ 18.5pF

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

6.4.1.2 Check Crystal ESR

Verifying that the crystal ESR falls within the suggested range ensures the reliable startup of the crystal circuit. The ESR is typically specified in the crystal data sheet. To confirm that the ESR falls within the suggested range, the negative resistance of the on-chip oscillator (R_{OSC}) is calculated with the equation below. Note that the external load capacitor values C_{L1} and C_{L2} must be derived before performing this calculation (see Calculating Load Capacitor Values). It is recommended that the negative resistance magnitude be greater than ~5x ESR to guarantee sufficient margin.

$$R_{osc} = \frac{G_M C_{X1_total} C_{X2_total}}{w^2 * \left(C_{X1_total} C_{X2_total} + C_{X1_total} C_{SHUNT} + C_{X2_total} C_{SHUNT} \right)^2 + (G_M C_{SHUNT})^2}$$

Where:

C_{X1_total} C_{X2_total} = total capacitance at X1/OSC and X2
C_{SHUNT} = parallel shunt capacitance from the crystal datasheet (typically 5–7pF)
G_M = Oscillator Transconductance from GMSL datasheet (typically 28mA/V)
w = angular frequency = 2π*25MHz

6.4.1.2.1 Example Crystal ESR Calculation

C_{X1_total} = 38pF, C_{X2_total} = 36pF, C_{SHUNT} = 7pF

$$R_{osc} = \frac{28mA/V * 38pF * 36pF}{(25MHz * 2\pi)^2 * \left(38pF * 36pF + 38pF * 7pF + 36pF * 7pF \right)^2 + (28mA/V * 7p)^2}$$

$$R_{osc} \sim 303\ ohms$$

As stated previously, the crystal ESR should be ~< R_{OSC} / 5. In this case, the ESR of the crystal employed in this application should be < 60Ω.

ADI Confidential for Review Under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers under NDA

6.4.1.3 Check Crystal Power Dissipation

Checking the crystal power dissipation helps to ensure the long-term reliability of the crystal. Exceeding the recommended drive level can cause accelerated aging and degrade reliability. Evaluating the drive level requires estimating the current through the crystal. Direct measurement of this parameter is difficult; however, it can be estimated by measuring the AC voltage at the X1/OSC pin and then calculating the current based on the estimated total capacitance between this pin and ground. A low capacitance probe must be used to avoid introducing error into the measurement. After estimating the current, the power can be calculated using the typical ESR value from the crystal data sheet.

Power dissipated by the crystal is:

$$P_{DRIVE} = I^2_{DRIVE(RMS)} * ESR$$

The current through C_1 is (note that C_1 is equivalent to C_{X1_total} in the previous examples):

$$I_{DRIVE(RMS)} = I_{C1(RMS)} = \left|\frac{V_{C1(RMS)}}{Z_{C1}}\right| = V_{C1(RMS)} * 2\pi f C_1$$

V_{C1} (voltage across C_1) is a sine wave, and $V_{rms} = \frac{V_{pp}}{2\sqrt{2}}$

The resulting current is $I_{DRIVE(RMS)} = \frac{V_{C1(P-P)}\pi f C_1}{\sqrt{2}}$

The dissipation is then estimated using the ESR specified in the crystal data sheet. The calculated drive level should be less than the maximum level specified in the crystal data sheet.

$$P_{DRIVE} = \frac{1}{2}(V_{C1(P-P)}\,\pi f C_1)^2 * ESR$$

6.4.1.3.1 Example Crystal Power Dissipation Calculation

800mVpp sine wave at 25MHz is measured at X1/OSC
ESR is specified at 45 ohms (typical value from crystal data sheet)
C_1 = 38pF

$$P_{DRIVE} = \frac{1}{2}(0.8V * \pi * 25MHz * 38pF)^2 * 45ohm = 128uW$$

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribution Only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

6.4.2 RCLKOUT Configuration

RCLKOUT (25 MHz, 25MHz/2, or 25MHz/4) configuration details are presented in Table 14.

Table 14. Configuration Registers for RCLKOUT

Register	Bitfield	Description
REG3	RCLKSEL[1:0]	Set the RCLKOUT frequency 00: 25MHz 01: 12.5MHz 10: 6.25MHz 11: Not supported in this mode (Reference PLL output)
REG3	RCLK_ALT[0]	Select which pin to output the RCLKOUT on.
REG6	RCLKEN[0]	Set to 1 to enable the RCLKOUT

6.4.2.1 Predefined DPLL Output

The configuration registers for the predefined DPLL frequency outputs are listed in Table 15. If REFGEN_PREDEF_FREQ_ALT is set low, 19.2, 27, 37.125, and 74.25MHz outputs are available; if REFGEN_PREDEF_FREQ_ALT is set high, 13.5, 24, 27, and 27MHz outputs are available.

Table 15. Configuration Registers for Predefined DPLL Frequencies

Register	Bitfield	Description
REF_VTG0	REFGEN_EN[0] REFGEN_RST[0]	Set REFGEN_EN = 0, set REFGEN_RST = 1 to reset the DPLL (can be skipped if DPLL has not been programmed)
REF_VTG0	REFGEN_PREDEF_FREQ_ALT[0] REFGEN_PREDEF_FREQ[1:0]	Select which pre-defined frequency to output: If REFGEN_PREDEF_FREQ_ALT[0] = 0 REFGEN_PREDEF_FREQ[1:0] = 00: 19.2MHz 01: 27MHz 10: 37.125MHz 11: 74.25MHz If REFGEN_PREDEF_FREQ_ALT[0] = 1 REFGEN_PREDEF_FREQ[1:0] = 00: 13.5MHz 01: 24MHz 10: 27MHz 11: 27MHz
REF_VTG0	REFGEN_EN[0] REFGEN_RST[0]	Set REFGEN_EN = 1 and REFGEN_RST = 0 to take DPP out of reset
REF_VTG0	REFGEN_LOCKED[0]	Read REFGEN_LOCKED to verify DPLL locked onto selected frequency.
REG3	RCLKSEL[1:0]	Set to 11 to set the RCLKOUT frequency to use the DPLL output
REG3	RCLK_ALT[0]	Select which pin to output the RCLKOUT on.
REG6	RCLKEN[0]	Set to 1 to enable the RCLKOUT

Maxim Integrated Confidential
Provided to GMSL2_Customers_Only
ADI Confidential for NDA Revision under NDA

6.4.2.2 Arbitrary DPLL Output

The DPLL can be programmed to arbitrary user-defined frequencies. A software tool is available to calculate the register writes necessary to produce the target DPLL output frequency. This software tool also produces the configuration script and theoretical frequency difference (target vs. actual output frequency). The configuration script follows the programming guide outlined in Table 16.

Note: Due to the finite precision of the algorithm, the DPLL circuit has finite precision and will generate a frequency output within ±3ppm of the programmed frequency.

Table 16. Configuration Registers for Arbitrary DPLL Frequencies

Register	Bitfield	Description
REF_VTG0	REFGEN_EN[0] REFGEN_PREDEF_FREQ[0] REFGEN_RST[0]	Set REFGEN_EN = 0, set REFGEN_PREDEF_FREQ = 0 to turn off the pre-defined DPLL option, set REFGEN_RST = 1 to reset the DPLL.
REF_VTG4 **REF_VTG5**	REFGEN_FB_FRACT_L[7:0] REFGEN_FB_FRACT_H[3:0]	Program with values given by the configuration tool
REF_VTG0	REFGEN_RST[0]	Set REFGEN_RST = 0 to take DPLL out of reset
DPLL_3 **DPLL_7** **DPLL_8** **DPLL_9** **DPLL_10** **DPLL_11**	Reference device-specific datasheet for bitfield names.	Program registers with values given by the configuration tool; this configures the PLL feedback loop gain and clock divider.
REF_VTG0	REFGEN_EN[0]	Set REFGEN_EN = 1 to enable DPLL
REF_VTG0	REFGEN_LOCKED[0]	Read REFGEN_LOCKED to verify DPLL locked onto selected frequency.
REG3	RCLKSEL[1:0]	Set to 11 to set the RCLKOUT frequency to use the DPLL output
REG3	RCLK_ALT[0]	Select which pin to output the RCLKOUT on.
REG6	RCLKEN[0]	Set to 1 to enable the RCLKOUT

6.4.2.3 Alternative RCLKOUT or DPLL_OUT Routing

The RCLKOUT/DPLL_OUT frequency output can be alternatively routed to a specified GPIO output. Table 17 contains configuration registers for routing RCLKOUT; register programming for DPLL routing is listed in Table 18 (predefined DPLL frequencies) and Table 19 (arbitrary DPLL frequencies).

Table 17. Alternate Configuration Registers for RCLKOUT

Register	Bitfield	Description
REG3	RCLKSEL[1:0]	Set the RCLKOUT frequency 00: 25MHz 01: 12.5MHz 10: 6.25MHz 11: Reference PLL output (don't select for this mode)
REF_VTG1	PCLKEN[0] RCLKEN_Y[0]	Set PCLKEN = 1 and set RCLKEN_Y = 1 to enable

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential, Provided to GMSL Customer Under NDA

REF_VTG1	PCLK_GPIO[4:0]	Select the desired MFP pin to output to: 00000: GPIO0 pin selected for PCLK output 00001: GPIO1 pin selected for PCLK output 00010: GPIO2 pin selected for PCLK output 00011: GPIO3 pin selected for PCLK output 00100: GPIO4 pin selected for PCLK output 00101: Not supported 00110: Not supported 00111: GPIO7 pin selected for PCLK output 01000: GPIO8 pin selected for PCLK output Others: Not supported

Table 18. Alternate Configuration Registers for Predefined DPLL Frequencies

Register	Bitfield	Description
REF_VTG0	REFGEN_EN[0] REFGEN_RST[0]	Set REFGEN_EN = 0, set REFGEN_RST = 1 to reset the DPLL (can be skipped if DPLL has not been programmed)
REF_VTG0	REFGEN_PREDEF_FREQ_ALT[0] REFGEN_PREDEF_FREQ[1:0]	Select which pre-defined frequency to output: If REFGEN_PREDEF_FREQ_ALT[0] = 0 REFGEN_PREDEF_FREQ[1:0] = 00: 19.2MHz 01: 27MHz 10: 37.125MHz 11: 74.25MHz If REFGEN_PREDEF_FREQ_ALT[0] = 1 REFGEN_PREDEF_FREQ[1:0] = 00: 13.5MHz 01: 24MHz 10: 27MHz 11: 27MHz
REF_VTG0	REFGEN_EN[0] REFGEN_RST[0]	Set REFGEN_EN = 1 and REFGEN_RST = 0 to take DPP out of reset
REF_VTG0	REFGEN_LOCKED[0]	Read REFGEN_LOCKED to verify DPLL locked onto selected frequency.
REG3	RCLKSEL[1:0]	Set to 11 to set the RCLKOUT frequency to use the DPLL output
REF_VTG1	PCLKEN[0]	Set PCLKEN = 1 to enable
REF_VTG1	PCLK_GPIO[4:0]	Select the desired MFP pin to output to: 00000: GPIO0 pin selected for PCLK output 00001: GPIO1 pin selected for PCLK output 00010: GPIO2 pin selected for PCLK output 00011: GPIO3 pin selected for PCLK output 00100: GPIO4 pin selected for PCLK output 00101: Not supported 00110: Not supported 00111: GPIO7 pin selected for PCLK output 01000: GPIO8 pin selected for PCLK output Others: Not supported

ADI Confidential for NDA
Maxim Integrated Confidential, Provided to GMSL2_Customer under NDA

Table 19. Alternate Configuration Registers for Arbitrary DPLL Frequencies

Register	Bitfield	Description
REF_VTG0	REFGEN_EN[0] REFGEN_RST[0]	Set REFGEN_EN = 0, set REFGEN_RST = 1 to reset the DPLL (can be skipped if DPLL has not been programmed)
REF_VTG0	REFGEN_PREDEF_FREQ_ALT[0] REFGEN_PREDEF_FREQ[1:0]	Select which pre-defined frequency to output: If REFGEN_PREDEF_FREQ_ALT[0] = 0 REFGEN_PREDEF_FREQ[1:0] = 00: 19.2MHz 01: 27MHz 10: 37.125MHz 11: 74.25MHz If REFGEN_PREDEF_FREQ_ALT[0] = 1 REFGEN_PREDEF_FREQ[1:0] = 00: 13.5MHz 01: 24MHz 10: 27MHz 11: 27MHz
REF_VTG0	REFGEN_EN[0] REFGEN_RST[0]	Set REFGEN_EN = 1 and REFGEN_RST = 0 to take DPP out of reset
REF_VTG0	REFGEN_LOCKED[0]	Read REFGEN_LOCKED to verify DPLL locked onto selected frequency.
REG3	RCLKSEL[1:0]	Set to 11 to set the RCLKOUT frequency to use the DPLL output
REF_VTG1	PCLKEN[0]	Set PCLKEN = 1 to enable
REF_VTG1	PCLK_GPIO[4:0]	Select the desired MFP pin to output to: 00000: GPIO0 pin selected for PCLK output 00001: GPIO1 pin selected for PCLK output 00010: GPIO2 pin selected for PCLK output 00011: GPIO3 pin selected for PCLK output 00100: GPIO4 pin selected for PCLK output 00101: Not supported 00110: Not supported 00111: GPIO7 pin selected for PCLK output 01000: GPIO8 pin selected for PCLK output Others: Not supported
REF_VTG1	PCLKEN[0]	Set PCLKEN = 1 to enable
REF_VTG1	PCLK_GPIO[4:0]	Select the desired MFP pin to output to: 00000: GPIO0 pin selected for PCLK output 00001: GPIO1 pin selected for PCLK output 00010: GPIO2 pin selected for PCLK output 00011: GPIO3 pin selected for PCLK output 00100: GPIO4 pin selected for PCLK output 00101: Not supported 00110: Not supported 00111: GPIO7 pin selected for PCLK output 01000: GPIO8 pin selected for PCLK output Others: Not supported

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

6.5 Applications and Examples

6.5.1 Crystal Frequency Reference

The typical frequency reference use case is pairing an external crystal with the on-chip oscillator as shown in *Figure 15*. If this method is used, a pair of external load capacitors is needed in addition to the crystal. The crystal's frequency precision and temperature stability must comply with the required values from the GMSL2 device's datasheet. The external load capacitor values, ESR, and drive level should be evaluated/checked as described above.

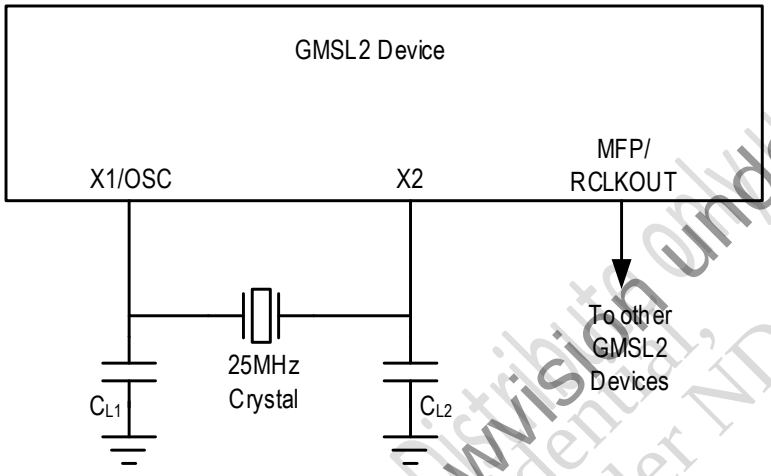


Figure 15. Crystal Reference

6.5.2 External Clock Source

An external clock source meeting the specified electrical characteristics can be connected directly to the X1/OSC pin as shown in Figure 16. The jitter and fall time of the external clock source must be within specified ranges to ensure good performance.

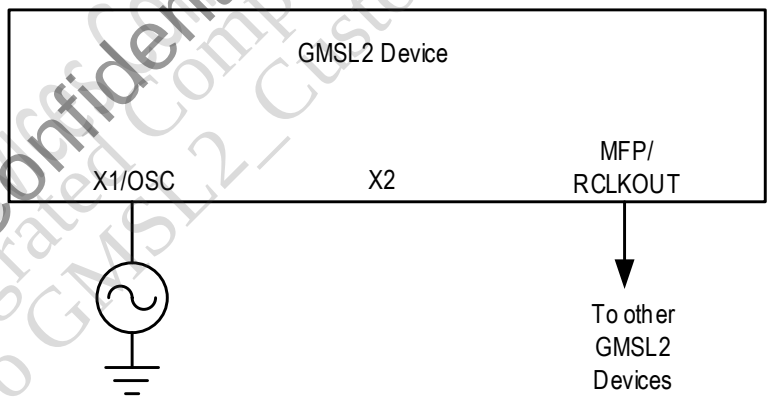


Figure 16. External Clock Source

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution Only Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

6.5.3 Using RCLKOUT to Drive the Frequency Reference Off-Chip

A common frequency reference can be used by multiple GMSL2 devices via the RCLKOUT MFP function as shown in *Figure 17*. The characteristics of the RCLKOUT reference are compliant with the external clock source specifications for other GMSL2 device(s). However, it is important to consider the potential accumulation of excessive jitter if the RCLKOUT reference is driven to multiple devices.

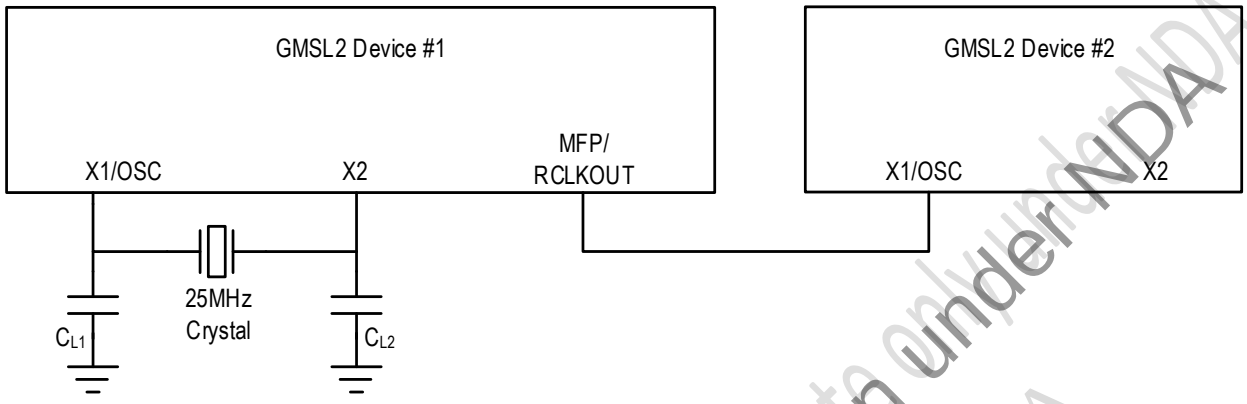


Figure 17. Using RCLKOUT to Share a Common Crystal

The RCLKOUT Configuration section provides additional details regarding the use and configuration of the RCLKOUT functions.

6.6 Feature Availability by Device Family

The frequency reference input is required and common to all GMSL2 devices. MAX9283/85 do not require a clock input as they operate in GMSL1 mode only. RCLKOUT and DPLL output support by part number is given in Table 20. Note that pin used for frequency output varies by device.

Table 20. Clock Reference Output Support by Device Family

Device Family	Frequency Reference Input	RCLKOUT	DPLL_OUT	RCLKOUT Pin	RCLKOUT Alt pin
HDMI Serializers	Yes	No	No	No	No
Advanced HDMI Serializers	Yes	No	No	No	No
DSI Serializers (except MAX9283/85)	Yes	Yes	No	MFP3	No
DSI Serializers (MAX9283/85)	No (GMSL1 only)	No	No	No	No
Advanced DSI Serializers	Yes	Yes	No	MFP3	No
CSI-2 Serializers	Yes	Yes	Yes	MFP4	MFP2
Advanced CSI-2 Serializers	Yes	Yes	No	MFP3	No
CSI-2 Camera Deserializers	Yes	No	No	No	No
CSI-2 Display Deserializers	Yes	No	No	No	No
OLDI Deserializers	Yes	No	No	No	No
eDP/DP Deserializers	Yes	No	No	No	No
CSI-2 Quad Deserializers	Yes	No	No	No	No

Maxim Confidential Distribute only under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA
Maxim Confidential – Not for Distribution – Revision under NDA

7 PRBS Testing

7.1 Overview

GMSL2 devices are equipped with internal pseudo-random binary sequence (PRBS) tests. PRBS tests are a type of built-in self-test (BIST) that enable the device to test and debug a system. Further, GMSL2 systems can use PRBS testing to determine the bit error rate (BER) of video/audio packets and of the GMSL link itself.

The forward channel PRBS uses a pattern generator in the serializer and a corresponding pattern checker in the deserializer. The serializer populates video or audio packets with PRBS data that is then checked by the deserializer. Some devices also support reverse channel audio PRBS. Here, the deserializer generates audio PRBS data to be decoded and checked by the serializer.

7.2 Operation

GMSL2 devices may offer up to three PRBS tests to evaluate supported channels. Refer to Feature Availability by Device Family.

- Forward Channel Video PRBS
- Forward Channel Audio PRBS
- Reverse Channel Audio PRBS

For GMSL2 camera parts that do not support audio, only the forward video PRBS test is available.

Note: GMSL2 devices have an error generator that can be used to verify that the PRBS is operational and recording errors properly. To verify the PRBS test, enable the ERRG during the PRBS test. Ensure that the PRBS checker is identifying expected errors. Refer to the *Error Generator (ERRG)* section for additional information.

The Video and Audio PRBS generator/checker blocks in GMSL2 serializers and deserializers are illustrated in Figure 18 and Figure 19, respectively.

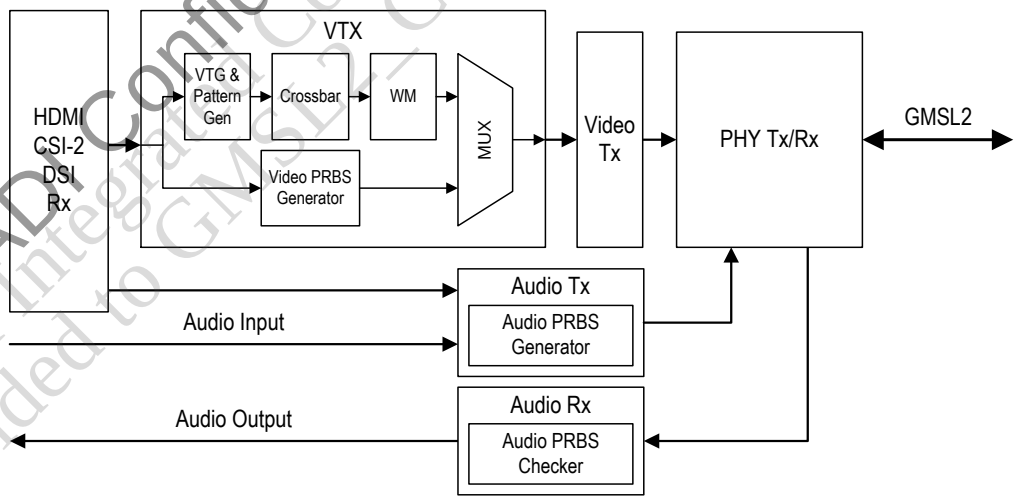


Figure 18. GMSL2 Serializer Video and Audio PRBS Generator/Checker Blocks

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2 Customer Under NDA

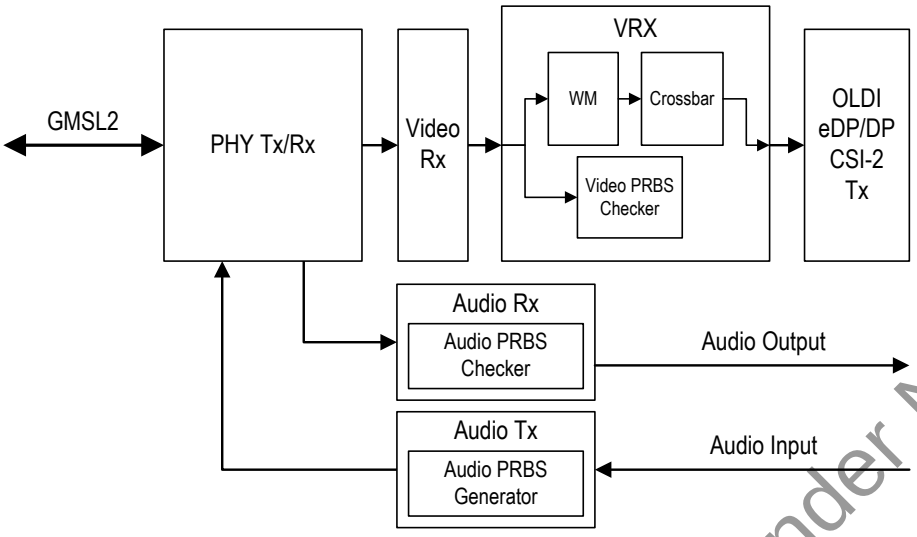


Figure 19. GMSL2 Deserializer Video and Audio PRBS Generator/Checker Blocks

7.2.1 Forward Channel Video PRBS

The forward channel video PRBS evaluates the video channels by testing the BER of video packets. This test is limited to the video channel(s), and the serial link is not affected. To enable the video PRBS test, the serializer must receive a valid clock from the video interface (i.e., HDMI, DSI, or CSI-2).

When the video PRBS test is enabled, the video packets in the serializer are replaced with PRBS data. The PRBS data is then sent to the PRBS checker in the deserializer via the serial link. Once received, the video packets are decoded and the PRBS data is output from the deserializer.

Video PRBS data is always 24 bits per pixels (bpp). RGB888 video has 24bpp (8 bits per color). But bits per pixel value varies with other video formats such as RAW, YUV. When bits per pixel value is different than 24bpp, it must be overridden to 24bpp with soft_bpp{x,y,z,u} and soft_bpp{x,y,z,u}_en before performing VPRBS testing. Refer to CSI-2 Serializers and Deserializers section for additional information.

7.2.2 Audio PRBS

Audio PRBS is available for supported devices (see Feature Availability by Device Family). To enable the audio PRBS, a valid clock must be provided. Clock sources include:

- Internal oscillator (selectable by register write)
- External clock provided via the I2S clock pin (SCK)

GMSL2 systems have forward and reverse audio functionality; separate audio PRBS tests are available for each channel. “Forward audio” refers to the forward channel direction: the serializer generates the audio PRBS data, and the deserializer decodes and checks the audio PRBS data. “Reverse audio” refers to the reverse channel direction: the deserializer generates the audio PRBS data, and the serializer decodes and checks the audio PRBS data.

Before enabling either audio PRBS test, the audio block’s Automatic Repeat Request (ARQ) must be disabled. If ARQ is enabled during a PRBS test, audio errors may not be flagged. The ARQ mechanism on the transmit side

Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA
Analog Devices Confidential, Provided to GMSL2 Customers Under NDA
ADI Confidential for New Version under NDA

automatically triggers the retransmission of packets if it does not receive confirmation of error-free transmission from the receive side. Retransmission continues until the error is corrected.

Note: Audio ARQ must be disabled to conduct Audio PRBS tests. Audio CRC can remain enabled. Reference the *CRC Error Detection and ARQ Error Correction* section for more information.

7.3 Configuration

7.3.1 Forward Channel Video PRBS

The video PRBS is used to evaluate and verify the video channel between the serializer and deserializer. The test requires a valid clock source. To begin testing, first enable the generator in the serializer (transmitter) by writing VID_PRBS_EN = 1, then enable the checker in the deserializer (receiver) by writing VPRBS_CHK_EN = 1. If the PRBS checker cannot synchronize with the incoming PRBS pattern, the VPRBS_FAIL register bit will be set. See Table 21 for full register configuration information.

The video PRBS test finds bit errors related to video packetization and decoding. The number of errors is stored in the VPRBS_ERR register (clears on read). The errors will be sent to the ERRB pin and pull ERRB low if the VPRBS_ERR_OEN register bit is enabled.

To stop the video PRBS test, first disable the checker in the receiver by writing VPRBS_CHK_EN = 0, then disable the generator in the transmitter by writing VPRBS_CHK_EN = 0.

GMSL2 camera devices are equipped with video pattern generator (VPG) that can be used as video source for the forward video PRBS test. Refer to Video Pattern Generator (VPG) section of the GMSL2 Users Guide.

The video PRBS test can be configured and enabled with the GMSL2 GUI tool. Ensure that a valid clock source is provided before enabling the test.

Note: The total serial link bandwidth is not completely consumed by the video channel. During video PRBS tests, bit errors unrelated to the video channel (e.g., link or audio channels) will not be flagged by the video PRBS error register.

A list of the video PRBS registers, bits, and their functions can be seen below in Table 21. Note that a GUI tool is also available for this configuration.

Table 21. Video PRBS Registers

Device	Bitfield	Description	Read/Write
Serializer	VID_PRBS_EN = 1	Enable bit for Video PRBS Generator	Write
Deserializer	VPRBS_CHK_EN = 1	Enable bit for Video PRBS Checker	Write
HDMI Serializer	SEL_EXT_PCLK = 0	Video PRBS clock selection (HDMI input)	Write
Deserializer	VPRBS_ERR	Video PRBS error count & read-on-clear register	Read
Deserializer	VPRBS_FAIL	Verify Video PRBS Checker synchronization status	Read

Table 22 contains an example of the register sequence necessary to enable video PRBS in a GMSL2 system comprising an HDMI serializer connected to a dual OLDI deserializer.

ADI Confidential for Review Only
Maxim Integrated Confidential, Provided Under NDA
Analog Devices Confidential, Distribution Only to GMSL 2 Customers Under NDA

Table 22. HDMI Serializer to OLDI Deserializer Video PRBS Test Sequence

Step	Device	Register Address	Bitfield (Device)	Description	Value
1	Serializer	0x0001	HDMI_AUTOS	Enable HDMI video	0x88 (Write)
2	Serializer	0x01C9	SEL_EXT_PCLK	Set HDMI video port as PRBS clock	0x00 (Write)
3	Serializer	0x01C9	(VTX) PCLKDET	Read PCLKDET to confirm HDMI data	0xE1 (Read)
4	Serializer	0x01E5	VID_PRBS_EN	Enable video PRBS	0x80 (Write)
5	Deserializer	0x01CD	VPRBS_CHK_EN	Enable video PRBS check	0x10 (Write)
6	Deserializer	0x01CC	VPRBS_ERR	Read video PRBS error count	Read
7	Deserializer	0x01CD	VPRBS_FAIL	Read video PRBS checker synchronization	0x20 (Read)

To disable video PRBS from the above example, first disable the video PRBS checker in the deserializer, then disable the video PRBS generator in the serializer.

While image sensor data is active and transmitting through the CSI-2 serializer and CSI-2 deserializer, if bits per pixel value is different than 24bpp, and heartbeat is disabled, a special video PRBS test sequence is required to avoid image sensor data corrupting the video PRBS data. Refer to CSI-2 Serializers and Deserializers section for details.

ADI Confidential for NewVision Under NDA
Analog Devices Confidential Distributed Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

7.3.2 Audio PRBS

The audio PRBS test finds bit errors related to audio packetization and decoding. The test requires a valid clock source and that the audio ARQ be disabled. To test the forward audio channel, first enable the generator in the serializer (transmitter) by writing PRBSEN_AUD = 1, then enable the checker in the deserializer (receiver) by writing APRBS_CHK_EN = 1. To test the reverse audio channel, enable the generator in the deserializer (transmitter) by writing PRBSEN_AUD = 1, then enable the checker in the serializer (receiver) by writing APRBS_CHK_EN = 1.

Verify that the audio PRBS test is running by reading the APRBS_VALID register bit on the PRBS receiver device. If the audio PRBS test is valid, the number of audio PRBS errors can be read from the APRBS_ERR register byte. The APRBS_ERR register clears upon read.

To disable the audio PRBS test, perform the following sequence in order: disable the checker in the receiver, disable the generator in the transmitter, and re-enable the ARQ for the audio block. All ARQ and CRC interrupts configured OEN = 1 should be cleared after the Audio PRBS test is completed. Alternatively, the device can undergo a power-on reset (POR), or power cycling event, to reset the interrupts.

Audio PRBS registers, bits, and their functions can be seen in Table 23 below.

Table 23. Audio PRBS Registers

Bitfield	Description	Read/Write
EN/ARQ_EN = 0	Disable Audio ARQ0	Write
PRBS_SEL = 0 (use audio clock SCK) PRBS_SEL = 1 (use internal oscillator)	Audio PRBS clock selection	Write
PRBSEN_AUD = 1	Enable bit for Audio PRBS Generator	Write
APRBS_CHK_EN = 1	Enable bit for Audio PRBS Checker	Write
APRBS_VALID	Verify Audio PRBS Checker synchronization status	Read
APRBS_ERR	Audio PRBS error count & clear-on-read register	Read

Note: The Audio ARQ bitfield may vary by device; refer to device datasheets for register information.

Note: Audio ARQ must be disabled to conduct Audio PRBS tests. Audio CRC can remain enabled. Reference the *CRC Error Detection and ARQ Error Correction* section for more information.

ADI Confidential for New vision under NDA
Analog Devices Confidential, Not to be distributed
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2 Customers Under NDA

7.3.3 Backward-Compatible GMSL1 PRBS Test

GMSL2 devices that are GMSL1 backward compatible can also perform GMSL1 PRBS tests. Both sides of the serial link must have identical BWS, DRS, HIBW, HIGHIMM, DBL, HVEN, and PRBS_TYPE settings. The backward-compatible GMSL2 device must also be configured to be able to communicate with the GMSL1 device via I2C or UART. Refer to the GMSL1 Legacy Mode (Backward Compatibility) section for details.

The PRBS_TYPE register determines which GMSL1 PRBS test will be run. GMSL1 devices may have limited PRBS testing capabilities; refer to device datasheets for information regarding available tests. The PRBS_TYPE setting must be matched on both devices prior to enabling PRBS.

The "legacy" PRBS test (PRBS_TYPE = 0) is similar to the GMSL2 video PRBS, while the PRBS_TYPE = 1 tests the serial link.

Note: Do not use PRBS_TYPE = 0 with DBL = 1 or PRBS_TYPE = 1 with HIBW = 1.

The procedure to enable GMSL1 PRBS tests is similar to the procedure for GMSL2 devices.

- Test procedure for PRBS_TYPE = 0
  - Set PRBSEN = 1 in the serializer first and then in deserializer. Communication with the remote side is possible in this mode. Errors are generated in the serializer via the error generator and read in the deserializer via the PRBS_ERR registers. The PRBS_ERR register is cleared upon read. PRBS_OK = 0 in this mode.
- Test procedure for PRBS_TYPE = 1
  - Set PRBSEN = 1 in the deserializer first and then in the serializer. Communication via the serial link is not possible for the duration of the PRBS test. The length of the PRBS test pattern is set with the PRBS_LEN register. Set the PRBS length and enable the error generator in the serializer before enabling PRBS. After the test finishes, read PRBS_OK = 1 and the PRBS_ERR register in the deserializer.

7.3.4 BER Testing using the GMSL2 Idle Link

The GMSL2 serial link itself can also be used as a pseudo-PRBS test. The packet-based transmission protocol of the GMSL2 link sends data at a fixed rate while it is active. If there is no video, control, or other data to send, the link transmits 'idle packets.' These idle packets are filled with random data and undergo the same serial link protocol encoding, scrambling, etc. processes of other data types. Analyzing the number of errors per unit of time while sending at a fixed rate provides a bit error ratio (BER), which can be used to determine the quality of the link.

See Table 24 for an example of the classification of the BER with respect to time without an observed error.

Table 24. Observation Time Required for 95% Confidence Interval of Different BER (@ 6Gbps)

BER	Time Interval Between Errors
10^{-6}	~500 microseconds
10^{-9}	~500 milliseconds
10^{-12}	~8.3 minutes
10^{-15}	~5.8 days
10^{-18}	~16 years

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential Distributed Only
Provided to GMSL2 Customer Under NDA

7.4 Feature Availability by Device Family

All GMSL2 devices support Video PRBS. Select GMSL2 devices support Audio PRBS. Reference Table 25 for feature availability by device family.

Table 25. PRBS Support by Device Family

Device Family	Video PRBS	Audio PRBS
HDMI Serializers	X	X
Advanced HDMI Serializers	X	X
DSI Serializers	X	X
Advanced DSI Serializers	X	X
CSI-2 Serializers	X	
Advanced CSI-2 Serializers	X	
CSI-2 Camera Deserializers	X	
CSI-2 Display Deserializers	X	X
OLDI Deserializers	X	X
eDP/DP Deserializers	X	X
CSI-2 Quad Deserializers	X	

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distributed Only Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

8 Link Integrity Tools

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

9 Software Algorithms for Link Integrity Tools

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Video Transmission

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

10 Video Basics

GMSL devices transmit digital video. Digital video transmissions consist of a series of specifically formatted video frames. The architecture of these frames, which borrows heavily from analog video, must be understood to ensure proper performance of the GMSL device(s) and the system as a whole. This section defines and reviews the core concepts of digital video, the terminology used throughout the GMSL2 User Guide, and the basic equations used to calculate video bandwidth.

10.1 Video Frame Architecture

The different regions of a video image are identified below in Figure 20. Note that there are variations in how these parameters are defined across the industry and that their usage may differ from analog video. The definitions below are used throughout this document and are applicable to all GMSL products. When designing systems, be aware that concepts and parameters presented here may not align with other industry definitions.

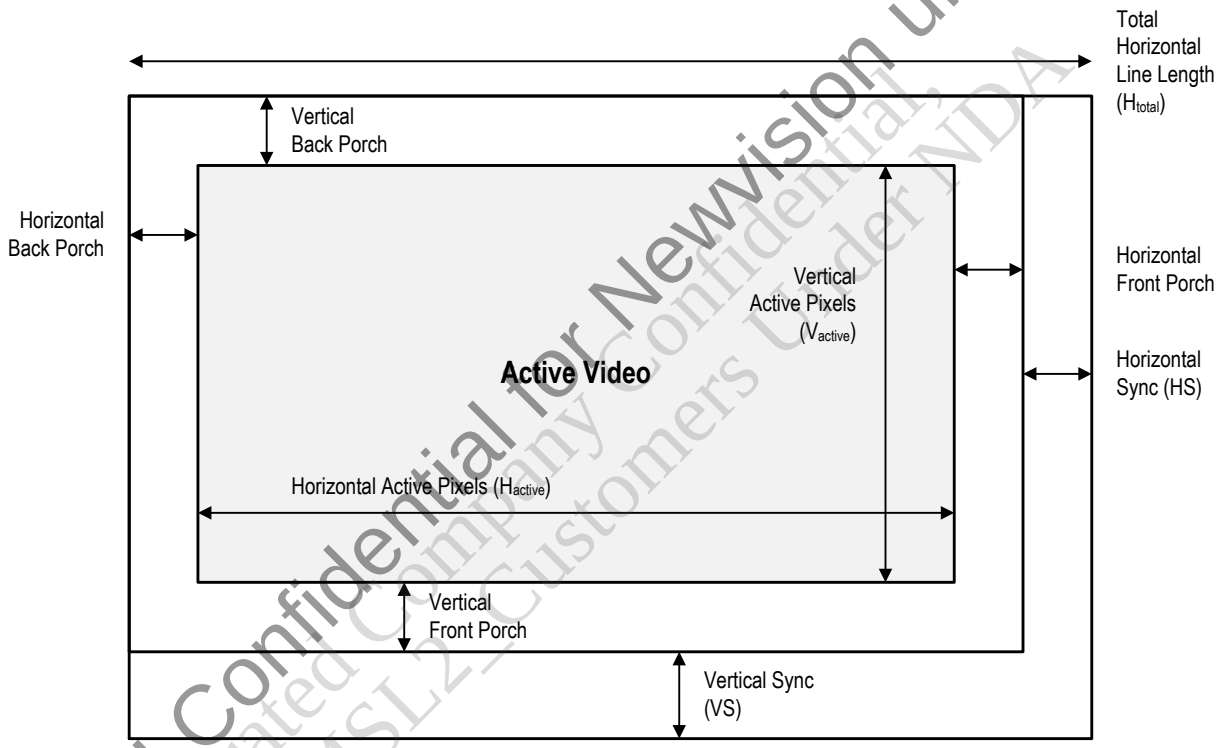


Figure 20. Video Frame Regions

The above diagram includes both the active video and blanking periods. The active video frame consists of horizontal lines scanned from left to right. The first pixel transmission begins in the top left corner of the frame. After the horizontal line has completed scanning, a new line is started directly below it, and the left-to-right scan pattern repeats. This process continues until the last line of the frame is completed, and the process restarts for a new frame. Note that the front porch and back porch locations are not intuitive. This naming convention comes from the parameters' location in time relative to the sync pulse (see Figure 21 below).

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

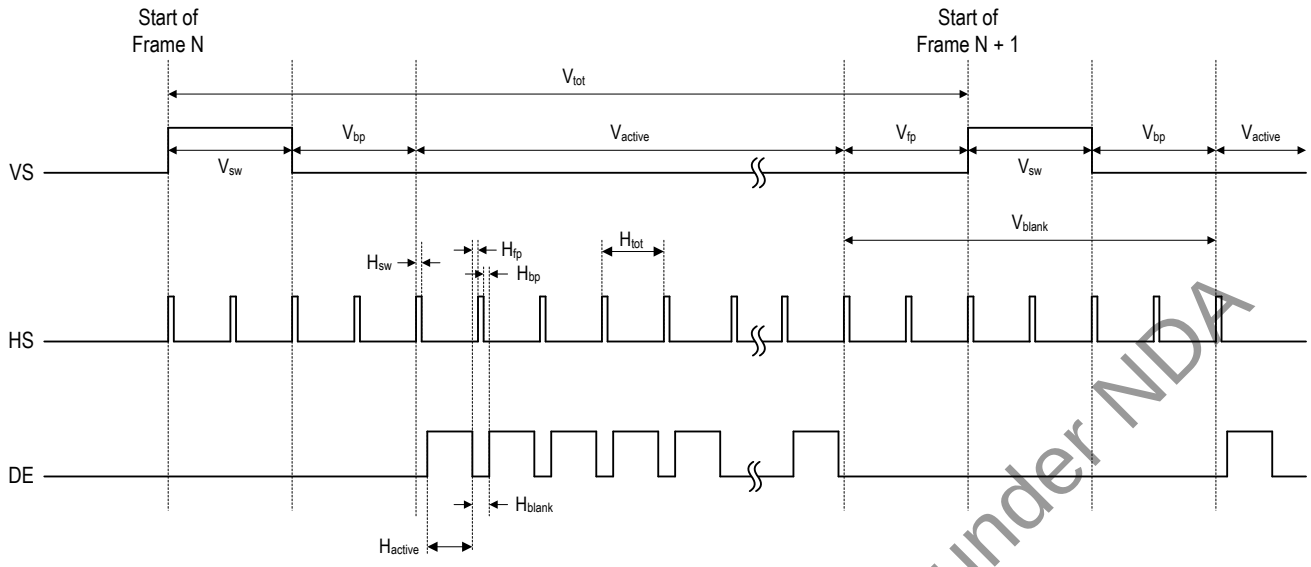


Figure 21. Sync Pulses: Data Enable and the Vertical and Horizontal Sync Signals

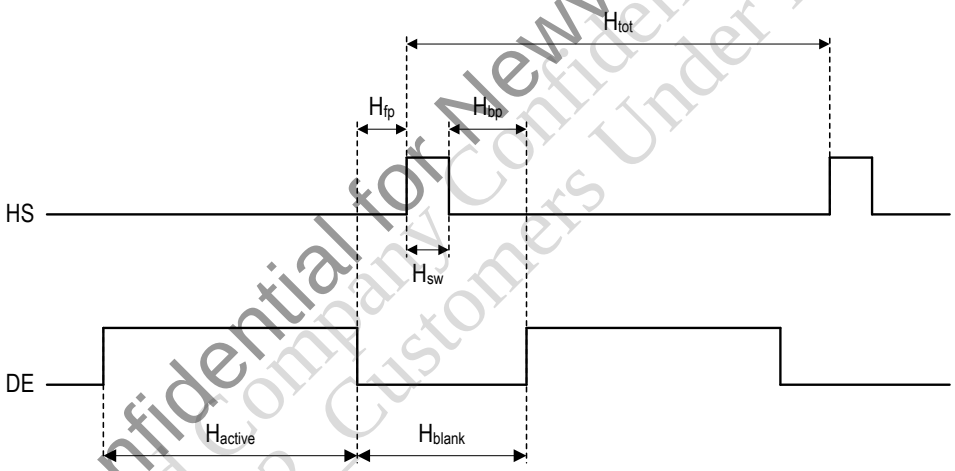


Figure 22. Horizontal Sync and Data Enable Signals

The diagram in Figure 21 shows the relationships between the Vertical Sync (VS), Horizontal Sync (HS), and Data Enable (DE) signals. Figure 22 exaggerates the horizontal blanking period to illustrate the relationship between the HS and DE signals as well as the video frame parameters.

Analog Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Distributed
Provided to GMSL2 Customers Under NDA

10.2 Equations

This section contains video timing and blanking equations and demonstrates how to calculate video bandwidth. Note that total link bandwidth available for video data is less than the link transfer speed due to link encoding overheard and side channel usage (e.g., main control channel, SPI, etc.). Also, the pixel clock and GMSL bandwidth must consider the total resolution of a video (i.e., video data plus blanking). Refer to the GMSL2 Link System Bandwidth section for further details.

10.2.1 Relationships Between Video Signals

Vertical Sync (VS)

- V_{SW} – Vertical Sync Width
- V_{FP} – Vertical Front Porch
- V_{BP} – Vertical Back Porch
- V_{active} – Vertical Active
- V_{tot} – Total Vertical Line Length
- V_{blank} – Vertical Blanking

Horizontal Sync (HS)

- H_{SW} – Horizontal Sync Width
- H_{FP} – Horizontal Front Porch
- H_{BP} – Horizontal Back Porch
- H_{active} – Horizontal Active
- H_{tot} – Total Horizontal Line Length
- H_{blank} – Horizontal Blanking

The video is active (i.e., the display time in which the video contains visible pixel data) during the horizontal active period. The horizontal blanking period separates the active video of each visible horizontal line. The horizontal sync (during the blanking period) indicates the start of a new horizontal line. The vertical sync signals have analogous functions to the horizontal counterparts; however, the vertical signals are used to indicate frame boundaries and are sometimes used to trigger other events (e.g., buffer resets) between frames. The vertical sync signal occurs during the vertical blanking period.

These concepts are illustrated in the above diagram (Figure 20). The active video (i.e., display time) is equivalent to the video screen that actively shows video data. The pixel data advances horizontally one pixel with each successive pixel clock period until the entire horizontal line is formed. The end of the horizontal line is designated by a horizontal sync pulse (HS) and signifies a shift to begin a new horizontal line directly below. This process continues until the entire frame is filled. When the entire frame is filled with horizontal lines, a vertical sync pulse (VS) indicates the end of the frame. The frame then refreshes and the process repeats. Note that cameras and SoCs often express resolution in terms of “active video” (i.e., viewable resolution). This designation ignores blanking. The pixel clock and GMSL bandwidth is dependent on the total resolution, however, and includes blanking.

ADI Confidential For New Version Under NDA
Analog Devices Confidential Information
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

10.2.2 Total Blanking Interval

$$Horizontal\ Blanking = (H.Front\ Porch) + (H.Sync\ Width) + (H.Back\ Porch)$$

$$Vertical\ Blanking = (V.Front\ Porch) + (V.Sync\ Width) + (V.Back\ Porch)$$

10.2.3 Total Line and Frame Period

$$H_{total} = Horizontal\ Active\ Pixels + Horizontal\ Blanking$$

$$V_{total} = Vertical\ Active\ Lines + Vertical\ Blanking$$

10.2.4 Pixel Clock (PCLK)

Calculating the pixel clock is necessary to determine if video transmissions are compatible with interface and GMSL2 link bandwidth availability.

$$PCLK = H_{total} * V_{total} * Framerate$$

10.2.4.1 Pixel Clock (Alternative Calculation)

In some situations, the specific blanking timings are not known. In these cases, an approximation of the blanking timing (called "Blanking Ratio" below) should be used to calculate an approximate PCLK value. Blanking ratios are typically in the range of 1.1–1.25; those values should be used for approximate calculations.

$$PCLK = (Active\ Width) * (Active\ Height) * (Blanking\ Ratio) * (Frame\ Rate)$$

10.2.4.2 Sample Calculation

$$1920\ h.pixels * 1080\ v.pixels * 1.2\ blanking\ ratio * 60\ Hz = 149MHz\ pclk$$

10.2.5 Calculating Total Video Bandwidth

$$Video\ BW = PCLK * bpp$$

Where bpp = bits per pixel. For RGB888 video this is 24 bits per pixel (8 bits per color). Note that the bits per pixel value varies with other video formats (e.g., RAW).

For example, a 1920x1080 60Hz 24-bit RGB display has a video bandwidth of ~3.56 Gbps.

The GMSL2 link has additional overhead that shares bandwidth with video data. Therefore, a 6Gbps video stream cannot be transferred on a 6Gbps GMSL2 link. Refer to the GMSL2 Link System Bandwidth section for further details.

ADI Confidential for New vision under NDA
Analog Devices Confidential – Distribution Only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

10.3 Definitions

This section contains definitions of basic video terminology.

10.3.1.1 Aspect Ratio

The ratio of the visible picture width to the height. Traditional televisions and monitors have an aspect ratio of 4:3 (1.33). The standard aspect ratio of modern computer monitors and automotive infotainment displays is 16:9 (1.78). Automotive instrument cluster displays are typically 8:3 (2.66), but custom aspect ratios are also common.

10.3.1.2 Back Porch

The area of a composite video signal defined as the time between the end of the sync signal and the start of the of active video.

10.3.1.3 Blanking Interval

There are horizontal and vertical blanking intervals. The horizontal blanking interval is the time period from the last active pixel in a line to the first active pixel in the following line. The vertical blanking interval is the time period from the last active pixel in a field or frame to the first active pixel in the following field or frame. The synchronizing signals occupy a portion of the blanking interval.

10.3.1.4 Color Bars

A standard video waveform used to test the calibration of a video system. It consists of a sequence of seven colored bars of a standard amplitude and size. The standard active-low color sequence is white, yellow, cyan, green, magenta, red, and blue. There are several amplitude standards, the most common being 75% amplitude (brightness) with 100% saturation (intensity of the color). GMSL2 devices can generate a color bar gradient pattern; refer to the Video Pattern Generator (VPG) section for details.

10.3.1.5 Data Enable

The signal that defines valid video data. When the Data Enable signal is off, the video data will be ignored. Note that cameras typically do not use DE signals, and instead use the HS and VS signals to indicate valid video data. Some display applications use only DE signals, while the HS and VS sync signals are ignored.

10.3.1.6 Fields and Frames

A frame is one complete scan of a picture. In interlaced scanning systems, a field is half of a frame; thus, two fields make a frame.

10.3.1.7 Frame Rate

See Vertical Frame Rate.

10.3.1.8 Front Porch

The area of a composite video waveform between the end of the active video and the leading edge of sync.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA

10.3.1.9 GMSL

Gigabit Multimedia Serial Link. GMSL is a proprietary serial link protocol used to transmit video, bi-directional audio, and bi-directional communication channel data over an automotive-grade physical interface.

10.3.1.10 Horizontal Blanking

The horizontal blanking interval is the time period from the last active pixel in a line to the first active pixel in the following line. Synchronizing signals occupy a portion of the blanking interval.

10.3.1.11 Horizontal Line Frequency

The inverse of the period of the total horizontal line time.

10.3.1.12 Horizontal Sync (HS)

The beginning of a horizontal line is indicated by a HS "pulse" during the horizontal blanking period. Sometimes, the Horizontal Sync is replaced by Data Enable.

10.3.1.13 Interlaced Scan

The process whereby each frame of a picture is created by first scanning half of the lines and then scanning the second set of lines. The second set of lines is interleaved between the first set to complete the picture. Each half is referred to as a field. Two fields make a frame. See also Progressive Scan.

10.3.1.14 Pixel

Picture element. A pixel is the smallest piece of display detail and each has a unique brightness and color. In a digital image, a pixel is an individual point in the image, represented by a certain number of bits to indicate the brightness.

10.3.1.15 Pixel Clock

The clock rate of pixels in a video stream. Pixel data advances at every clock edge.

10.3.1.16 Progressive Scan

The process whereby a picture is created by scanning all the lines of a frame in successive passes. Contrast with Interlaced Scan. The process of converting from interlaced to progressive scan is called "line doubling."

10.3.1.17 Refresh Rate

See Vertical Frame Rate.

10.3.1.18 Sync Signals/Pulses

Sync signals, also known as sync pulses, are timing pulses in video signals that are used by video processing or display devices to synchronize the horizontal and vertical portions of the display. They include *Horizontal Sync (HS)* and *Vertical Sync (VS)*. These signals typically occur during the blanking period(s).

10.3.1.19 Vertical Blanking

The vertical blanking interval is the time period from the last active pixel in a field or frame to the first active pixel in the following field or frame. Synchronizing signals occupy a portion of the blanking interval.

ADI Confidential for NewVision under NDA
Maxim Confidential Distribution only under NDA
Provided to GMSL2_Customers Under NDA
Integrated Company Confidential

10.3.1.20 Vertical Field Frequency

The inverse of the time (or period) to produce one field of video (half of a frame). In NTSC, it is 59.94Hz.

10.3.1.21 Vertical Frame Rate

The inverse of the time (or period) to produce one frame of video. Also called "refresh rate" or "vertical refresh rate."

10.3.1.22 Vertical Sync (VS)

The beginning of a frame is indicated by a VS “pulse” during the vertical blanking period in which data enable is also turned off.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

11 Camera Basics

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

12 Video Pipes

12.1 Overview

All GMSL2 devices use a common set of video data processing blocks to transfer video data across the serial link. This section describes the video input and output blocks and explains how data flows through the “video pipe” structures. Video pipes are used to route video data through the GMSL2 devices and contain several important video data processing features, including:

- Test features: VPG, VTG, and PRBS
- Color mapping tools: Video Crossbar and Color Lookup Tables (LUT)
- Authentication features: Watermark and HDCP
- HS and VS monitoring functions

12.2 Operation

The GMSL2 video channel is designed to transmit 8- to 24-bit video data. Video data is received by the serializer at the input video interface (e.g., HDMI), converted to parallel video data, and transmitted through the serializer video pipes to the GMSL packetizer. The data is then packetized and transmitted by the GMSL PHY across the serial link. The connected deserializer receives the data on the GMSL PHY, and the data is de-packetized into parallel video data. The parallel video data is transmitted through the video pipes to the output video interface (e.g., OLDI). Within the GMSL parts, the parallel video data consists of the pixel data and HS, VS, and DE sync signals. Figure 23 shows a functional block diagram of the video processing blocks within a GMSL system.

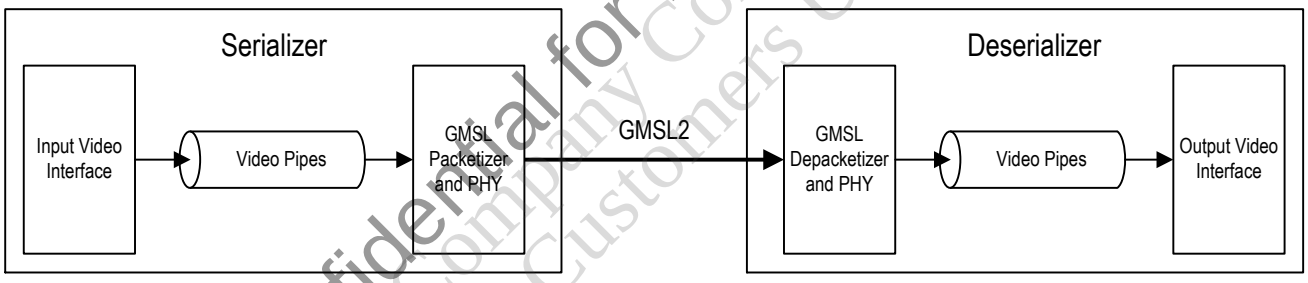


Figure 23. Video Pipes in a GMSL2 System

ADI Confidential for New Vision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

12.2.1 Video Packetizer (Serializer)

The video packetizer receives video data from the crossbar output when video transmission is enabled and the link is locked. Upon receiving video data, the packetizer prepares the video data to be transmitted throughout the serial link system.

The packetizer detects the presence of the PCLK and measures the frequency once it has stabilized. The measured PCLK value is transmitted to the deserializer via info frame packets for clock regeneration. The PCLK is monitored for maximum threshold and flag violations that result in video blanking. PCLK drift detection (i.e., ±1.25% drift without spread tracking and ±0.25% drift when tracking ±0.5% input spread at PCLK) reports errors to the DRIFT_ERR register and restarts the sub-system.

Note: The maximum PCLK value varies by part number. Refer to device-specific datasheet for more information.

The video is packed according to the selected mode and bits per pixel (bpp) setting to create video packets. These packets fill the transmit FIFO and a transmission request is sent to the scheduler. The video data is then transmitted to the serial link with optional video line CRC (enabled by default) or video pixel CRC (disabled by default). Reference the CRC Error Detection and ARQ Error Correction section for additional information and configuration details.

Note: The transmit FIFO has a level warning, auto-priority increment, and overflow detection.

12.2.2 Video Depacketizer (Deserializer)

The transmitted video packets are received by the GMSL PHY and sent to the depacketizer to be decoded. The data is then written to the video clock regeneration FIFO in a burst according to the selected encoding mode, bpp setting, and packet type. The clock regeneration module receives the decoded video from the link clock domain and streams out the video using the PCLK regenerated with the DPLL and the clock control loop.

Note: The number of video Rx blocks per deserializer varies by part number. Refer to device-specific datasheet for more information.

12.2.2.1 Video Packet Detector

The video packet detector observes the output of the packet receiver on the deserializer side and measures the density of video packets within the total link bandwidth to determine if video is running on the link. Using the 150MHz oscillator, the total valid video packets inside the link will be measured; packet detection will be set if the measured PCLK is larger than 5.3MHz. As there are 36 pixels in each packet, packet throughput of ~146 kilo packets per second is set as threshold.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribution Limited
Maxim Integrated Company Confidential, Customers Under NDA
Provided to CAVIUM_

12.3 Block Descriptions

The following sections briefly describe the blocks within the video pipes. More information can be found in the dedicated sections.

12.3.1 GMSL2 Serializers

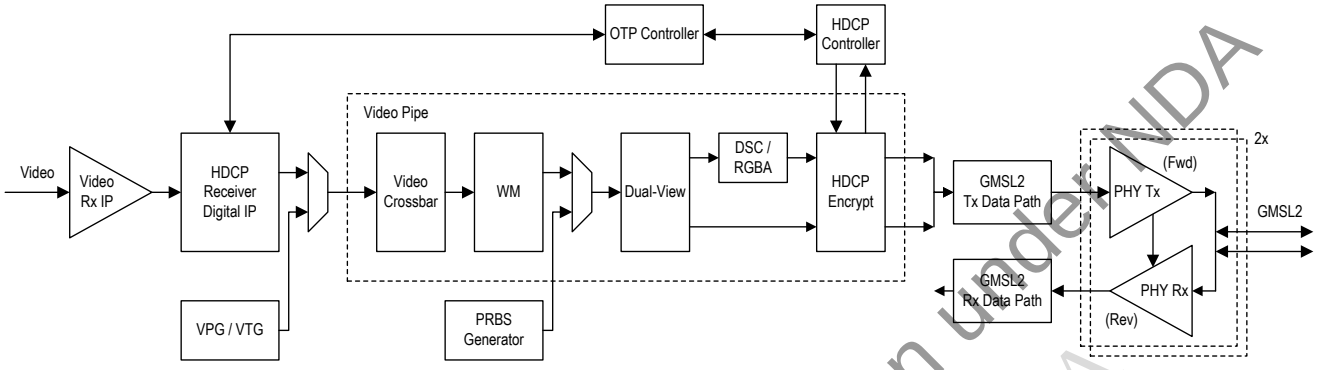


Figure 24. GMSL2 Serializer Video Pipe Structure

12.3.2 GMSL2 Deserializers

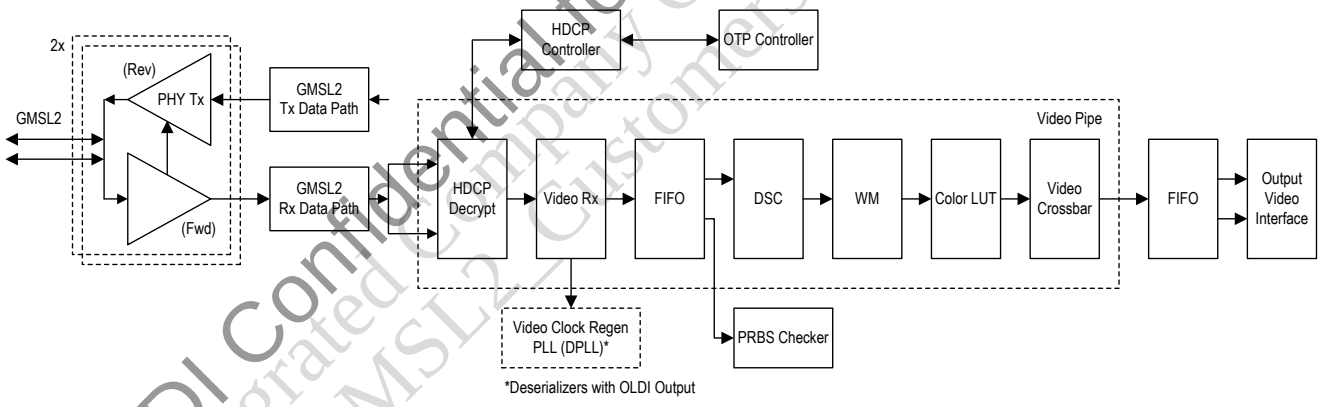


Figure 25. GMSL2 Deserializer Video Pipe Structure

12.3.3 Video Timing Generator (VTG)

The Video Timing Generator (VTG) enables the creation or modification of the HS, VS, and DE signals.

ADI Confidential for Newvision under NDA
Analog Devices Initial Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

12.3.4 Video Pattern Generator (VPG)

The Video Pattern Generator (VPG) creates either a checkerboard or gradient pattern with programable parameters. These patterns can be used to replace the incoming video or in conjunction with the VTG to create a video pattern when no video is present on the serializer input.

12.3.5 Watermark (WM)

Serializers contain a watermark generator and detector block in video pipe X. Deserializers contain a watermark generator and detector block in video pipe X. Reference the Watermarking section for additional information.

12.3.6 Video PRBS Tx/Rx

Serializer video pipes contain a video PRBS generator that sends a PRBS pattern within the video packets sent across the link; the PRBS pattern is checked by the video PRBS checker in the deserializer video pipe. Reference the PRBS Testing section for additional information.

12.3.7 Dual-View Control

Dual-View enables incoming video to be split into two video streams.

12.3.8 DSC Compression

On DSC-capable serializers, the Display Stream Compression (DSC) engine compresses the video data from 24 bits per pixel to 8 bits per pixel, providing a 3x reduction in bandwidth utilization across the serial link. On DSC-capable deserializers, the DSC decompression engine restores the video data to an uncompressed 24 bits per pixel.

12.3.9 HDCP

Serializers with HDCP support integrate an HDCP encryption block within the video pipe for encryption of video content. Deserializers with HDCP support have an HDCP decryption block in the video pipes for decryption of HDCP-encrypted video content. Note that both GMSL2 devices in the serial link system must support HDCP to enable the feature.

12.3.10 Video Crossbar

The Video Crossbar is a mux that allows any video input bit to be mapped to any output bit using the configuration registers. This is used to remap/reorder pixel components.

12.3.11 Color Lookup Table (LUT)

The Color Lookup Table (LUT) allows any color in RGB domain to be mapped to any other RGB color. This can be used for generating color filters or gamma correction maps.

12.3.12 HS and VS Monitoring

For debug purposes, HS and VS can be measured via dedicated GPIO pins. This function must be enabled for each signal individually. Refer to device-specific datasheets for details regarding the GPIO pin used for each monitor function and the location of the associated enable bit(s).

ADI Confidential for New Vision under NDA
Analog Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

12.4 Configuration

The video Tx/Rx blocks have the following programmable modes of operation:

- Heartbeat Mode On: Enables video clock regeneration (required for OLDI output)
- Heartbeat Mode Off: Disables video clock regeneration (CSI-2 and eDP/DP output)
- GMSL2 Stream ID Select
- GMSL2 PHY-Level Distribution of Video Streams

Refer to interface-specific sections for complete configuration details.

12.4.1 Heartbeat Mode On

Default setting. No additional configuration is required.

12.4.2 Heartbeat Mode Off

For applications where PCLK regeneration is not required, it is suggested to completely turn off data transmission during blanking periods. This eliminates video bandwidth consumption during blanking.

Note: Heartbeat mode should only be disabled when the deserializer device has CSI-2 or eDP/DP output.

Serializer:

- Disable heartbeat packet transmission: LIM_HEART = 1'b1

Deserializer:

- Disable the packet sequence number checker: SEQ_MISS_EN = 1'b0
- Disable the packet detector: DIS_PKT_DET = 1'b1

Note: If the GMSL2 device has multiple video pipes, the registers above are set individually for each video pipe.

12.4.3 GMSL2 Stream ID Select

Each active video pipe in the serializer transmits data with a unique STREAM_ID assigned with the TX_STR_SEL register. On the deserializer side, each video pipe has a configurable STR_SEL register that selects the STREAM_ID to be received from the serializer. Video streams can be duplicated by programming multiple deserializer video pipes to the same STR_SEL setting.

Note: The GMSL2 video STREAM_ID is only related to the GMSL2 protocol. It is distinct from the virtual channels of MIPI CSI-2 and the STREAM_ID of the HDMI or eDP/DP.

12.4.4 GMSL2 PHY-Level Distribution of Video Streams

Select GMSL2 serializer devices have two GMSL PHYs (i.e., PHY A and PHY B). Video pipe output can be configured to be sent from PHY A, PHY B, or both using the GMSL2 packet protocol scheduler. This is set with the TX_SPLT_MASK_A_VIDEO_X/Y and TX_SPLT_MASK_B_VIDEO_X/Y registers. For each serializer video pipe, these registers define the destination deserializer device(s).

ADI Confidential for NewVision under NDA
Analog Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

12.5 Status and Error Bits

The following register bits provide status and error information related to video pipe operation in serial link systems.

12.5.1 Deserializer Status and Error Bits

VID_LOCK (register VIDEO_RX8) — Indicates that a valid video stream is received for the selected channel ID.

Note: For OLDI deserializers, VID_LOCK also indicates that the valid video stream is properly recovered by the LVDS to drive the display. For all other deserializers, VID_LOCK = 1 when the video is received regardless of the status of the output interface.

VID_PKT_DET (register VIDEO_RX8) — Indicates that a stream of video packet for the selected stream ID is detected by the deserializer.

12.5.2 Serializer Status and Error Bits

PCLKDET (register VIDEO_TX2) — Status registers for video pipe X, Y, Z, and U. Normal operation reads 1 if the pipe is utilized.

OVERFLOW (register VIDEO_TX2) — Reports if video bandwidth is exceeded.

DRIFT_ERR (register VIDEO_TX2) — Reports a drift error caused by instability of the video PCLK frequency.

12.6 Feature Availability by Device Family

All GMSL2 devices have video pipes. Device-specific feature availability is presented in Table 26.

Table 26. Video Pipe Feature Support by Device Family

Device Family	Video Pipes	DSC	RGBA	Heartbeat Modes	Video Crossbar	HDCP	GMSL1 Mode
HDMI Serializers	2			X	X	X	
Advanced HDMI Serializers	2	X	X	X	X	X	X
DSI Serializers	4			X	X	X	X
Advanced DSI Serializers	4			X	X	X	X
CSI-2 Serializers	4			X	X		X
Advanced CSI-2 Serializers	4			X	X		X
CSI-2 Camera Deserializers	4	X	X	X	X		X
CSI-2 Display Deserializers	4			X	X	X	X
OLDI Deserializers	1			X	X	X	
eDP/DP Deserializers	4	X		X		X	
CSI-2 Quad Deserializers	8			X	X		X

ADI Confidential for Newvision under NDA
Material Confidential Distribute only to Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

13 Dual-View

13.1 Overview

Dual-view is a configurable feature included with HDMI and DSI GMSL2 serializers that enables a single serializer to support two different video streams. A serializer with dual-view enabled can receive a video stream comprising two videos combined into a single frame. This combined frame is called a superframe. The dual-view block takes the incoming superframe video stream, splits it into two symmetric or asymmetric video streams according to the enabled mode, and outputs the video onto two independent GMSL stream IDs. The output video streams can then be routed to one or more deserializers depending on application.

In the example below, a superframe of 3840x1200 @ 60Hz is created in the SoC by combining two videos side-by-side into a single frame. The superframe is output from the SoC and received by the serializer. The serializer, with symmetric dual-view enabled, splits the superframe into two separate video streams marked as 'Video A' and 'Video B' below in *Figure 26*. Each of these streams is 1920x1200 @ 60Hz.

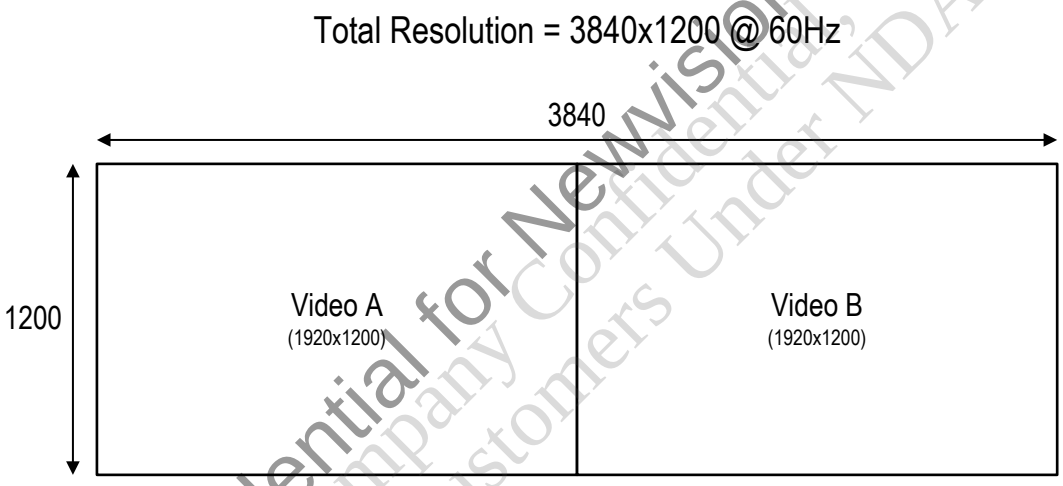


Figure 26. Example of Side-by-Side Symmetric Dual-View

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

13.2 Modes of Operation

Dual-view modes of operation are categorized as either symmetric or asymmetric. Asymmetric dual-view (sub-frames do not have equal video timing) is only supported by the Advanced DSI serializers; operation and configuration details are discussed in the DSI Serializers section.

13.2.1 Symmetric Dual-View Modes

Symmetric dual-view modes require that constituent videos (i.e., the sub-frames) of the incoming superframe have identical timing (i.e., active resolution, blanking interval, and frame rate). The incoming superframe video on the serializer input (from the SoC) is split into two separate sub-frames by the dual-view block and output by the serializer. Three types of symmetric dual-view superframes are supported:

- Pixel-interleaved dual-view mode.
- Side-by-side dual-view mode.
- Line-interleaved dual-view mode.

The choice of dual-view mode is dependent on the superframe timing as output by the SoC and received by the serializer. In each mode, the serializer should be configured to match the superframe that was created by the SoC. When configured, the superframe is split into two symmetrical video frames by the dual-view block of the serializer and output onto two independent GMSL stream IDs. *Figure 27* shows the timings for each sub-frame, Video A and Video B.

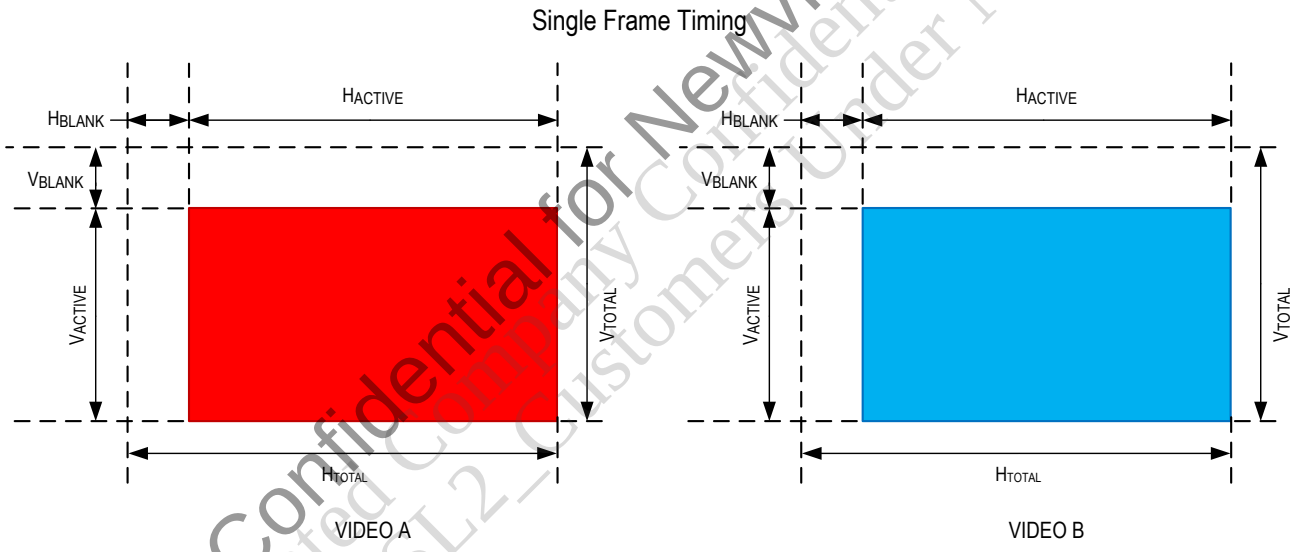


Figure 27. Single Frame Timing

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

13.2.1.1 Pixel-Interleaved Dual-View Mode

Pixel-interleaved mode (also known as column-interleaved mode) separates the video stream by assigning every odd pixel to one stream and every even pixel to another stream (Figure 28). This includes the horizontal blanking, which is split in half and symmetrically distributed to the two video streams. Vertical blanking timing is maintained from superframe to each sub-frame video stream. The PCLK is divided by two for each sub-frame.

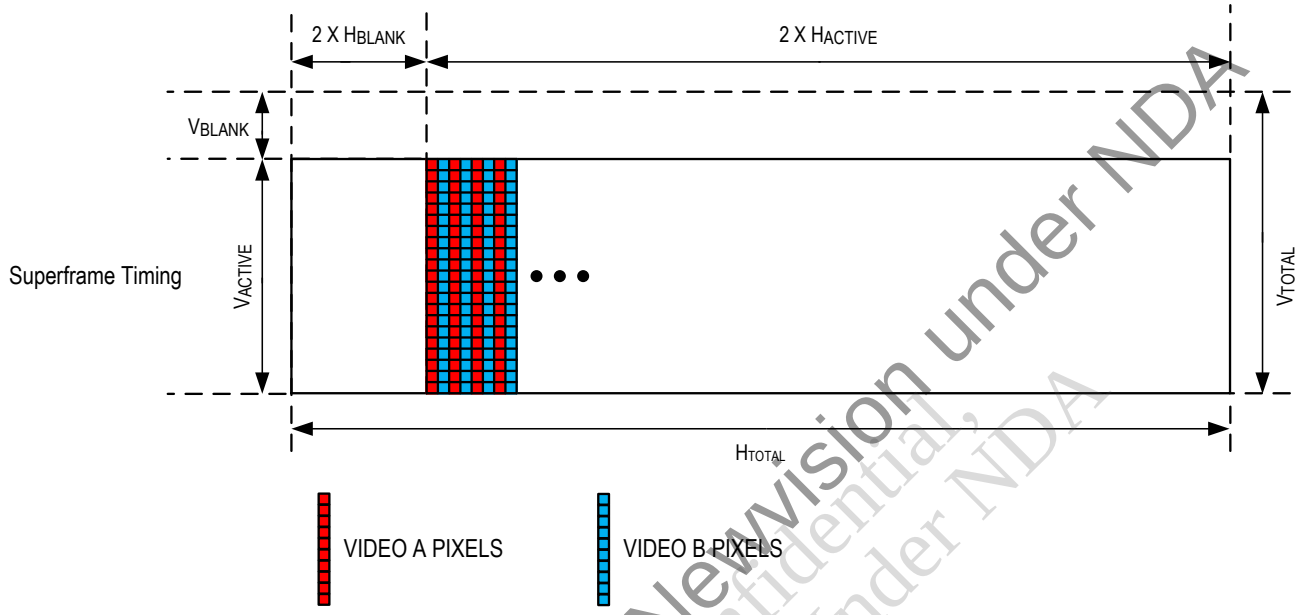


Figure 28. Pixel-Interleaved Dual-View Mode

ADI Confidential for NewVision under NDA
Analog Devices Confidential,
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

13.2.1.2 Side-by-Side Dual-View Mode

Side-by-side mode is used to split the superframe in half by creating sub-frames from the left and right halves of the superframe as shown in *Figure 29*. Side-by-side dual-view mode works by first converting the superframe data into pixel-interleaved data. Then, the pixel-interleaved data is separated by assigning every odd pixel to one stream and every even pixel to another stream. When side-by-side dual-view mode is enabled, the dual-view block measures the length of the superframe line that contains the side-by-side image format. Assuming that the pixel-length of the line is 2N, the pixels are reordered in the following order:

0,N,1,N+1,……..N-2,2N-2,N-1,2N-1

Here, the odd pixels are sent to one video stream and even are send to another stream.

Horizontal blanking is split symmetrically between streams, and vertical timing is maintained from superframe to the two sub-frame video streams. The PCLK is divided by two for each sub-frame.

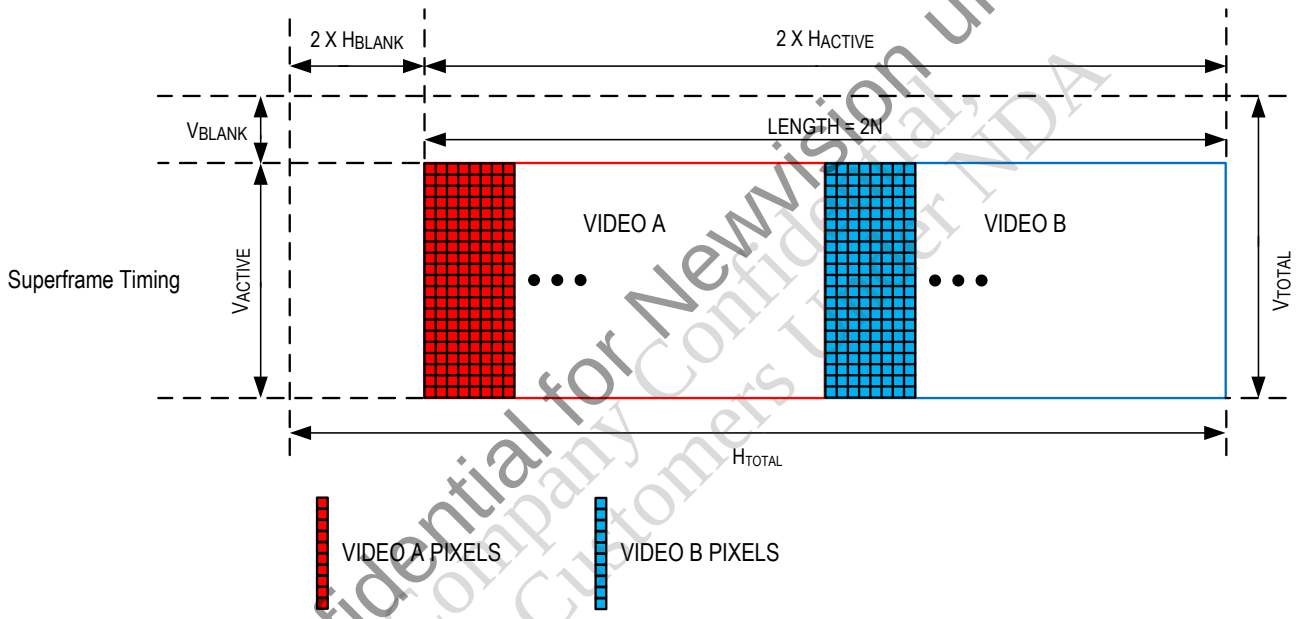


Figure 29. Side-by-Side Dual-View Mode

ADI Confidential for Newvision under NDA
Analog Devices, Inc. Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

13.2.1.3 Line-Interleaved Dual-View Mode

In line-interleaved mode, the incoming data is split horizontally line-by-line (Figure 30). Odd lines (beginning with the first active line of a frame) contain Video A; even lines contain Video B. When line-interleaved mode is enabled, odd lines are split into one video stream and even into the other. Horizontal blanking timing is maintained from superframe to sub-frame, but vertical timing is symmetrically split between streams. The PCLK is divided by two for the sub-frames.

Note: Not all devices support line-interleaved dual-view mode; refer to *Feature Availability by Device Family* for more information.

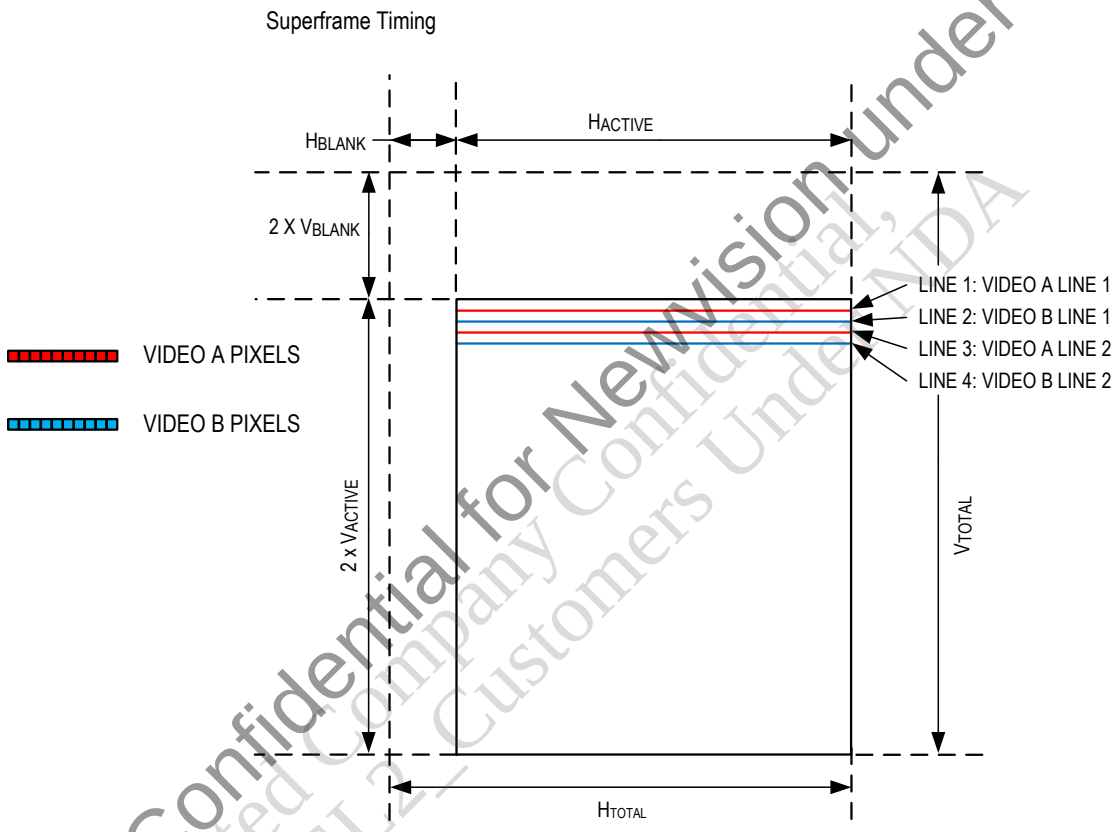


Figure 30. Line-Interleaved Dual-View Mode

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

13.2.2 Asymmetric Dual-View Mode

Asymmetric dual-view mode, in contrast to symmetric dual-view modes, processes a superframe comprising two images of different resolutions (Figure 31). The smaller image embeds uniformly padded dummy lines to match the vertical active period of the larger image. The padding must consider the FIFO overflow requirements in the asymmetric engine and the capabilities of the SoC providing the superframe. When the superframe is split into sub-frames by the asymmetric dual-view engine, the dummy lines are removed, and the original video resolution is restored. The serializer then transmits the sub-frames separately using two GMSL stream IDs, which can be transmitted via a single or two independent GMSL2 links (when splitter mode is used). The two individual video streams have user-defined timings to maintain the native output video timings (including PCLK frequency and accurate blanking values).

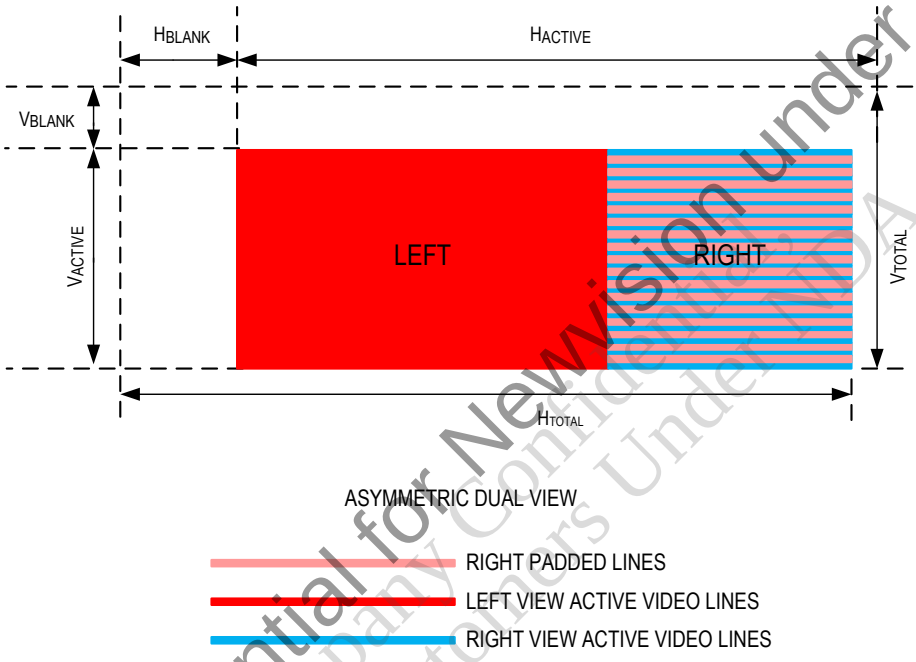


Figure 31. Asymmetric Dual-View

ADI Confidential for Newvision under NDA
Analog Devices (Maxim Integrated Company Confidential) Provided to GMSL2_ Customers Under NDA

In asymmetric dual-view mode, the timings of the left sub-frame, right sub-frame, and superframe are programmed into the dual-view engine along with the padding ratio. The dual-view block uses the left-sub-frame as the first video stream; the right sub-frame, with the padding lines removed, is output as the second video stream (Figure 32).

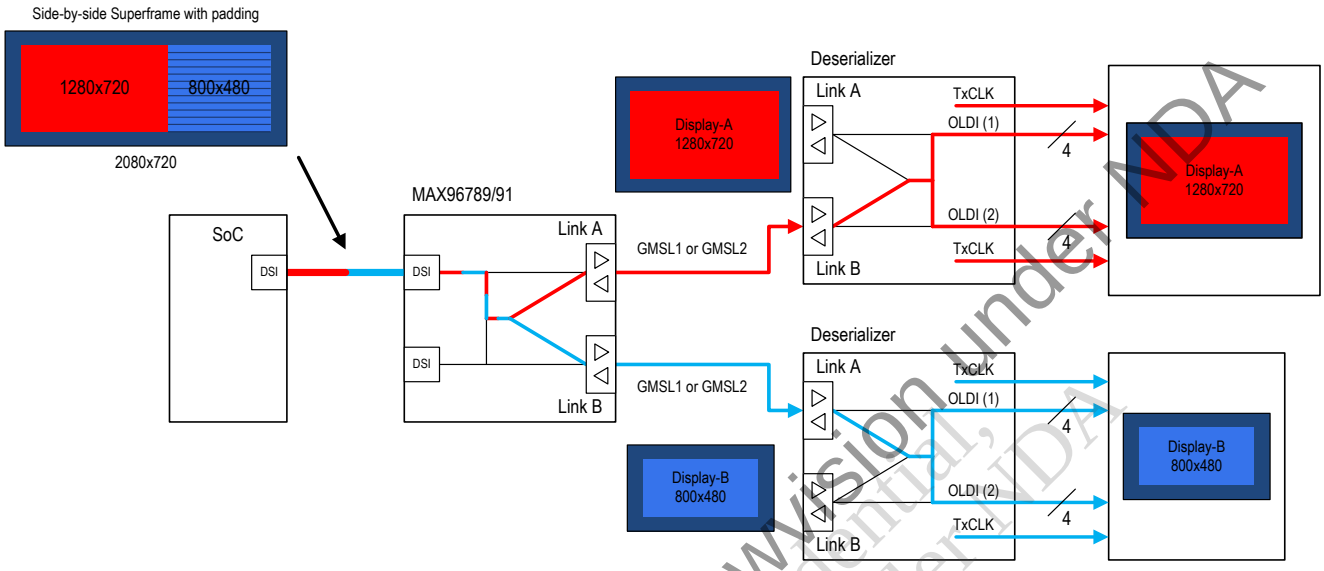


Figure 32. Asymmetric Dual-View Example

The video source generating the superframe must meet the following requirements for uniformly padded mode to be used. Reference Figure 33 for steps 1–5.

Note: The following terminology is interchangeable: Display 1/Display A and Display 2/Display B.

- The superframe is a single video stream that includes both videos: display 1 (left frame) and display 2 (right frame). The superframe is transmitted over a single video input (single high-speed DSI packet).
- The smaller frame in height must be on the right side of the superframe.
- The smaller frame (display 2) is padded with dummy lines to match the vertical height of display 1.
  - The "stuffed" (dummy) lines can be anything (e.g., empty or blank data).
- Display 2 should be padded uniformly and on a periodic basis to avoid internal memory overflow.
  - Predefined vertical ratios (display 1 to display 2) are user-selectable:
    - 1:1, 2:1, 3:1, 3:2, 4:1, 4:3, 5:2, 5:3, 8:5, 2:1 inverted, and the programmable look up table.
  - The vertical ratio is the relationship of the total active vertical lines of the larger frame to the smaller frame.
  - For applications that require padding not available by the predefined rations, the programmable Asymmetric Look Up Table (LUT) can be used to support any vertical ration. Note that the LUT requires a significant amount of register writes relative to the predefined ratios.
- The first active lines of the two videos are top aligned. The videos are attached side-by-side.

Analog Devices Confidential
Maxim Integrated Company Confidential,
Provided to GMSL Customers Under NDA
ADI Confidential for New Vision under NDA

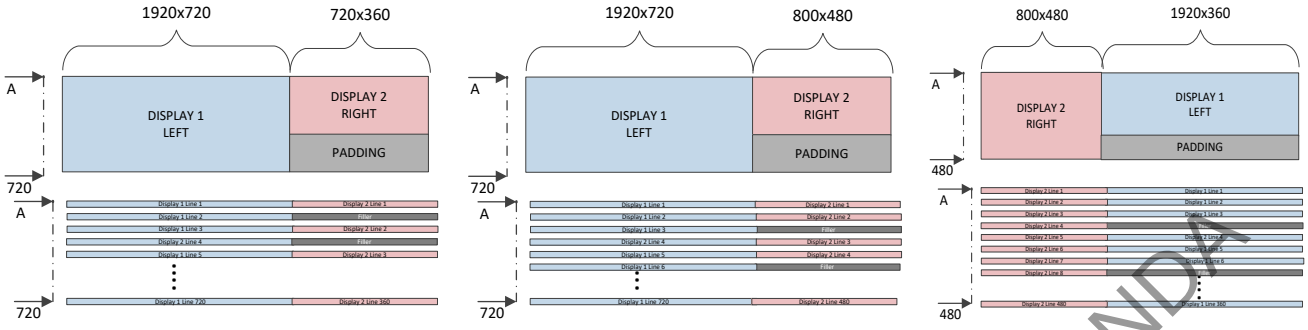


Figure 33. Predefined Ratios and Line Placement for Superframe Padding

- The PCLK frequencies of the superframe, display 1, and display 2 are individually determined by their respective resolutions (including blanking).
- The active video lines of the superframe includes both the left and right frames.
- The total height of the superframe should match the total height of the larger frame (Display 1).

$$VerticalSuperframe_{active} + VSF_{BP} + VSF_{FP} = VDisplay1_{ACT} + V1_{BP} + V1_{FP}$$

- The total vertical ratio (including blanking) of Displays 1 and 2 should match their total vertical active ratio:

$$\frac{Vtotal1}{Vtotal2} = \frac{Vactive1}{Vactive2}$$

- Asymmetric splitting can occur on either MIPI DSI port DSI0 or DSI1, but the video must be routed internally to pipe X (via register write).

Note: If the SoC does not have a uniform padding requirement, either of the videos may have padded lines where required. The registers ASYM_DV_LUT_A (LUT for display A) and ASYM_DV_LUT (LUT for display B) should be programmed to mark the lines to be removed from the output video.

ADI Confidential for New Version under NDA
Analog Devices Confidential, Distribution Prohibited
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

13.3 Configuration

Note: Dual-view configuration registers must be written in the order presented in the tables.

13.3.1 Pixel-Interleaved Dual-View

Enable dual-view mode by setting DV_EN = 1 in register DV0. Set DV_SPL = 1 to split the received superframe into sub-frames that are sent to video pipe X and video pipe Y. Odd pixels (starting with the first pixel of a line) will output to pipe X; even pixels will output to pipe Y. In register DV2, enable video pipe X by setting VID_EN_X = 1 and enable video pipe Y by setting VID_EN_Y = 1.

The stream ID of the pipe X video stream is set with TX_STR_SEL in register VIDEO_X TX3. If the device is in splitter mode (i.e., connected to two deserializers), prevent the video in pipe X from being sent to one or both SIOx serial link outputs by enabling TX_SPLT_MASK_A and/or TX_SPLT_MASK_B. Repeat these steps for pipe Y. This configuration option is typically used to save link bandwidth. If dual-view is being used in combination with splitter mode, these registers are used to ensure that correct signal routing is established (e.g., one video is sent to the deserializer on SIOA and the other video is only sent to the deserializer on SIOB).

The configuration process for enabling pixel-interleaved dual-view mode is presented in *Table 27*.

Table 27. Pixel-Interleaved Dual-View Configuration Registers

Step	Description	Register Name	Register Write
1	Enable dual-view block	DV0	DV_EN = 1
2	Split the video into two streams, outputting on pipe X and pipe Y	DV0	DV_SPL = 1
3	Select which pipe receives even/odd pixels	DV0	DV_SWP_AB = 0 sends odd pixels to pipe X and even pixels to pipe Y DV_SWP_AB = 1 sends even pixels to pipe X and odd pixels to pipe Y
4	Enable video stream on pipe X (default)	DV2	VID_EN_X = 1
5	Enable video stream on pipe Y	DV2	VID_EN_Y = 1
6	Assign pipe X stream ID and select which PHY to send video	VIDEO_X TX3	set TX_STR_SEL, TX_SPLT_MASK_A and TX_SPLT_MASK_B
7	Assign pipe Y stream ID and select which PHY to send video	VIDEO_Y TX3	set TX_STR_SEL, TX_SPLT_MASK_A and TX_SPLT_MASK_B

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution Prohibited
Maxim Integrated Company Confidential, Customer Under NDA
Provided to GMSL2_Customer Under NDA

13.3.2 Side-by-Side Dual-View

Enable dual-view mode by setting DV_EN = 1 in register DV0 and set DV_CONV = 1 to enable side-by-side to pixel-interleaved conversion. Set DV_SPL = 1 to split the received superframe into sub-frames that are sent to video pipe X and video pipe Y. Left-side pixels will output to pipe X; right-side pixels will output to pipe Y. To switch the output orientation, set DV_SWP_AB = 1 to output right-side pixels to pipe X and left-side pixels to pipe Y. In register DV2, enable video pipe X by setting VID_EN_X = 1 and enable video pipe Y by setting VID_EN_Y = 1.

The stream ID of the pipe X video stream is set with TX_STR_SEL in register VIDEO_X TX3. If the device is in splitter mode (i.e., connected to two deserializers), prevent the video in pipe X from being sent to one or both SIOx serial link outputs by enabling TX_SPLT_MASK_A and/or TX_SPLT_MASK_B. Repeat these steps for pipe Y. This configuration option is typically used to save link bandwidth. If dual-view is being used in combination with splitter mode, these registers are used to ensure that correct signal routing is established (e.g., one video is sent to the deserializer on SIOA and the other video is only sent to the deserializer on SIOB).

The configuration process for enabling side-by-side dual-view mode is presented in Table 28.

Table 28. Side-by-Side Dual-View Configuration Registers

Step	Description	Register Name	Register Write
1	Enable dual-view block	DV0	DV_EN = 1
2	Convert side-by-side to pixel interleaved video stream	DV0	DV_CONV = 1
3	Split the video into two streams, outputting on pipe X and pipe Y	DV0	DV_SPL = 1
4	Select which pipe receives even/odd pixels	DV0	DV_SWP_AB = 0 sends odd pixels to pipe X and even pixels to pipe Y DV_SWP_AB = 1 sends even pixels to pipe X and odd pixels to pipe Y
5	Enable video stream on pipe X (default)	DV2	VID_EN_X = 1
6	Enable video stream on pipe Y	DV2	VID_EN_Y = 1
7	Assign pipe X stream ID and select which PHY to send video	VIDEO_X TX3	set TX_STR_SEL, TX_SPLT_MASK_A and TX_SPLT_MASK_B
8	Assign pipe Y stream ID and select which PHY to send video	VIDEO_Y TX3	set TX_STR_SEL, TX_SPLT_MASK_A and TX_SPLT_MASK_B

ADI Confidential for Newvision under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customer Under NDA

13.3.3 Line-Interleaved Dual-View

Configuring line-interleaved dual-view requires configuring the dual-view block as well as programming the Video Timing Generator (VTG). The line-interleaved dual-view block divides the horizontal and vertical timings by two, so the VTG is required to double the horizontal timings before the dual-view block to preserve the original horizontal timing of the input. This is shown in Figure 34.

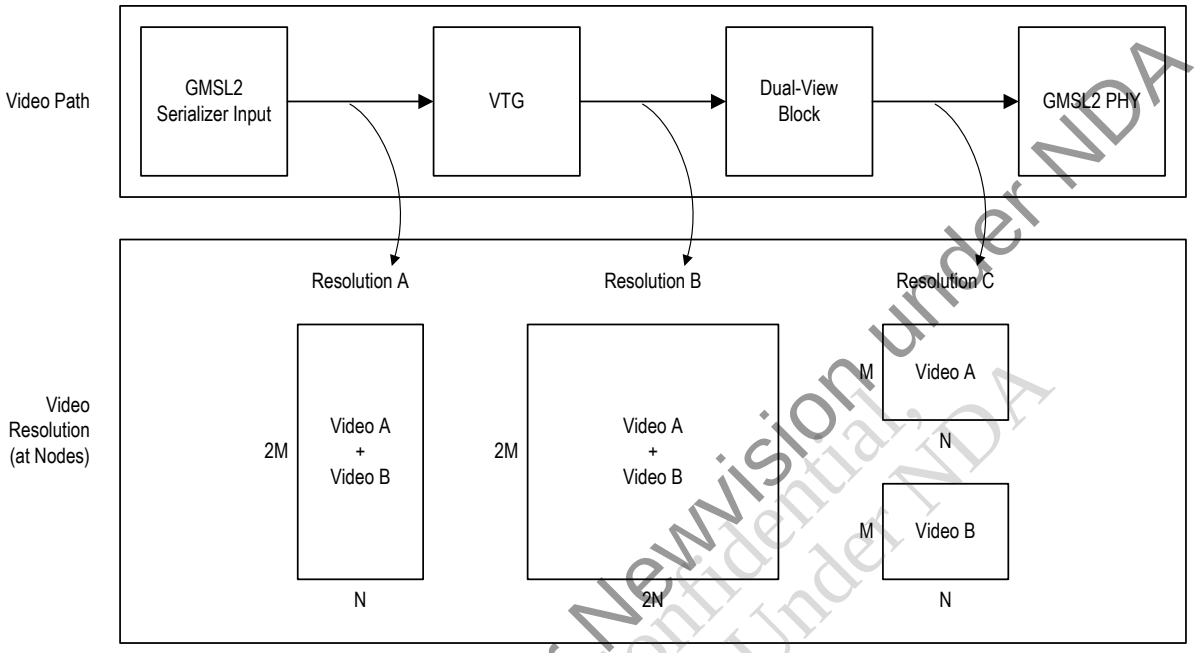


Figure 34. Video Timing Along the Video Path Through the Serializer

Enable line-interleaved dual-view mode by setting LINE_ALT = 1 in register DV0. Set DV_SPL = 1 to split the received superframe into sub-frames that are sent to video pipe X and video pipe Y. Odd lines (starting with the first line) will output to pipe X; even lines will output to pipe Y. To swap the between video pipe X and video pipe Y, set DV_SWP_AB = 1. In register DV2, enable video pipe X by setting VID_EN_X = 1 and enable video pipe Y by setting VID_EN_Y = 1.

The stream ID of the pipe X video stream is set with TX_STR_SEL in register VIDEO_X TX3. If the device is in splitter mode (i.e., connected to two deserializers), prevent the video in pipe X from being sent to one or both SIOx serial link outputs by enabling TX_SPLT_MASK_A and/or TX_SPLT_MASK_B. Repeat these steps for pipe Y. This configuration option is typically used to save link bandwidth. If dual-view is being used in combination with splitter mode, these registers are used to ensure that correct signal routing is established (e.g., one video is sent to the deserializer on SIOA and the other video is only sent to the deserializer on SIOB).

As noted previously, when using line interleaved mode, the serializer's Video Timing Generator (VTG) must be used to reconstruct the sync timings of the incoming video stream before it is sent to the dual-view block. A tool is available to create the correct VTG settings for a given superframe timing. These settings are also defined in the table below. For additional VTG operation and programming details, see the *Video Timing Generator (VTG)* section.

Finally, in serializer register DV0, set DV_EN = 1 to enable dual-view.

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

Note: The system is designed for positive VS polarity. If the VS polarity is negative, refer to configuration instructions in the *Line-Interleaved Dual-View with Negative Polarity* section below.

The below register writes are defined for a superframe with video timing:

- Horizontal total pixels = Htot
- Horizontal active pixels = Ha
- Horizontal front porch = Hfp
- Horizontal sync width = Hsw
- Horizontal back porch = Hbp
- Vertical total pixels = Vtot
- Vertical active pixels = Va
- Vertical front porch = Vfp
- Vertical sync width = Vsw
- Vertical back porch = Vbp

The configuration process for enabling line-interleaved dual-view mode is presented in Table 29.

Table 29. Line-Interleaved Dual-View Configuration Registers

Step	Description	Register Name	Register Write
1	Enable Line-interleaved dual-view mode	DV0	LINE_ALT = 1
2	Split the video into 2 streams, outputting on pipe X and pipe Y	DV0	DV_SPL = 1
3	Select which pipe receives even/odd pixels	DV0	DV_SWP_AB = 0 sends odd lines to pipe X and even pixels to pipe Y DV_SWP_AB = 1 sends even lines to pipe X and odd pixels to pipe Y
4	Enable video stream on pipe X (default)	DV2	VID_EN_X = 1
5	Enable video stream on pipe Y	DV2	VID_EN_Y = 1
6	Assign pipe X stream ID and select which PHY to send video	VIDEO_X TX3	set TX_STR_SEL, TX_SPLT_MASK_A and TX_SPLT_MASK_B
7	Assign pipe Y stream ID and select which PHY to send video	VIDEO_Y TX3	set TX_STR_SEL, TX_SPLT_MASK_A and TX_SPLT_MASK_B
8	**VTG**: Enable VTG, set to VS one-trigger mode, and generate DE, HS, and VS signals	VTX0	VTG_MODE[1:0] = 01 GEN_DE = 1 GEN_HS = 1 GEN_VS = 1
9	**VTG**: Trigger VS on rising edge	VTX1	VS_TRIG = 1
10	**VTG**: Time between VS edge of superframe and VTX VS generation. Used as a buffer for making sure first line of superframe is written to the memory before starting to read. (1 line + 40 pixels)	VTX2 - VTX4	VS_DLY = Htot + 40

ADI Confidential for NewVision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

11	**VTG**: VS high time of the superframe in terms of superframe pixels	VTX5 - VTX7	VS_HIGH = Vsw * Htot
12	**VTG**: VS low time of the superframe in terms of superframe pixels	VTX8 - VTX10	VS_LOW = (Vtot - Vsw) * Htot
13	**VTG**: Time between VS edge of superframe and first VTX HS generation. Must be equal to VS_DLY to keep VS and HS aligned.	VTX11 - VTX13	V2H = Htot + 40
14	**VTG**: Double HS high time in terms of superframe pixels	VTX14 - VTX15	HS_HIGH = 2 * Hsw
15	**VTG**: Double HS low time in terms of superframe pixels	VTX16 - VTX17	HS_LOW = 2 * (Htot - Hsw)
16	**VTG**: Half the number of HS pulses in the superframe	VTX18 - VTX19	HS_CNT = Vtot / 2
17	**VTG**: Time between VS rising edge of superframe and first VTG DE generation. This value is equal to VS_DLY + vertical sync width plus back porch time + two horizontal sync width plus horizontal back porch time	VTX20 - VTX22	V2D = (Htot + 40) + Htot * (Vsw + Vbp) + 2 * (Hsw + Hbp)
18	**VTG**: Two times the DE high time in a superframe line in terms of superframe pixels	VTX23 - VTX24	DE_HIGH = 2 * Ha
19	**VTG**: Two times the DE low time in a superframe line in terms of superframe pixels	VTX25 - VTX26	DE_LOW = 2 * (Hfp + Hsw + Hbp)
20	**VTG**: Half the number of active lines in the superframe	VTX27 - VTX28	DE_CNT = Vactive / 2
21	Enable dual-view block	DV0	DV_EN = 1

ADI Confidential for Newvision under NDA
Analog Devices Confidential Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

13.3.3.1 Line-Interleaved Dual-View with Negative Polarity

The system is designed for positive VS polarity. If the VS polarity is negative, invert VS using the CROSS_25 register. Then, perform the following sequence to start the system properly:

```
#Method
#program the configuration
#turn off video tx to link
0x80,0x01A2,0x8A

#start HDMI video
#wait for HDMI_SCDT,CKDT,PCLKDET to be 1

#selextpclk=1
0x80,0x01C9,0xF1

#force vs=0, de=0
0x80,0x01F2,0x3A
0x80,0x01F1,0x39
0x80,0x2A0C,0x02

#turn off dual-view
0x80,0x01A0,0x26

#selextpclk=0
0x80,0x01C9,0xE1

#release force vs,de
0x80,0x01F1,0x19
0x80,0x01F2,0x1A
0x80,0x2A0C,0x00

#turn on dual-view
0x80,0x01A0,0x27

#turn on video tx to the link
0x80,0x01A2,0xEA
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

13.3.4 Asymmetric Dual-View

Use the GUI configuration tool to generate the register settings for asymmetric dual-view mode.

13.3.4.1 Asymmetric Dual-View VS Shifting

The location of the VS signal in the output video can be changed for individual videos using the ASYM_READ_A and ASYM_READ_B retiming registers by exchanging lines between the VFP and VFB.

Note: Asymmetric dual-view VS shifting is available only to Advanced DSI Serializers.

13.4 Status and Debug Registers

If the system does not perform as expected, analyze the video source output using video analyzer equipment to ensure that it satisfies the exact timing requirements.

13.4.1.1 Symmetric Dual-View

The key status registers associated with symmetric dual-view mode are:

- DV_LOCK (register DV0) shows whether the dual-view engine is locked or not.
- PCLKDET (register VIDEO_TX2) registers for pipe X and pipe Y should both be 1 if the dual-view engine is running and sending video to the two corresponding pipes.
- VID_LOCK (register VIDEO_RX8) register of the deserializer expecting the video should be 1 after the serializer starts sending the video streams through the GMSL2 link.

If the above registers are not in the expected states, check the TX_SPLT_MASK_A and TX_SPLT_MASK_B bits of register CFGV_VIDEO_TX3 to determine whether the routing to the GMSL links are correct or not. Check that both video pipes used are enabled (VID_EN_X = VID_EN_Y = 1). Ensure that the TX_STR_SEL bits in the serializer register CFGV_VIDEO_TX3 and the STR_SEL bits in the deserializer register CFGH_VIDEO_RX0 indicate that the serializer and deserializer stream IDs match.

13.4.1.2 Asymmetric Dual-View

The key status registers associated with asymmetric dual-view mode are:

- PCLKDET (register VIDEO_TX2) registers for pipe X and pipe Y should both be 1 if the dual-view engine is running and sending video to the two corresponding pipes.
- VID_LOCK (register VIDEO_RX8) register of the deserializer expecting the video should be 1 after the serializer starts sending the video streams through the GMSL2 link.
- ASYM_OVERFLOW_A, ASYM_OVERFLOW_B, ASYM_UNDERFLOW_A, ASYM_UNDERFLOW_B (register ASYM27) should be 0 during normal operation of the asymmetric dual-view engine. If any of these bits are high, ensure that:
  - The video input timing matches the timing information provided to the asymmetric dual-view engine.
  - The display timings of the individual displays are suitable for asymmetric dual-view operation.

Analog Devices Confidential for New Version under NDA
Maxim Integrated Confidential, Company Confidential, Provided to CMLS Customers Under NDA

13.5 Feature Availability by Device Family

Dual-view is supported by GMSL2 HDMI and DSI serializers. Mode support varies by device family (Table 30).

Table 30. Dual-View Support by Device Family

Device Family	Pixel-Interleaved Dual-View	Side-by-Side Dual-View	Line-Interleaved Dual-View	Asymmetric Dual-View
HDMI Serializers	X	X	X	
Advanced HDMI Serializers	X	X		
DSI Serializers	X	X	X	
Advanced DSI Serializers	X	X	X	X
CSI-2 Serializers				
Advanced CSI-2 Serializers				
CSI-2 Camera Deserializers				
CSI-2 Display Deserializers				
OLDI Deserializers				
eDP/DP Deserializers				
CSI-2 Quad Deserializers				

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

14 Display Stream Compression (DSC)

14.1 Overview

The Display Stream Compression (DSC) feature in GMSL2 devices implements a video compression algorithm compliant with the VESA DSC v1.1 specification. This feature provides a fixed 3:1, visually-lossless compression of the video stream transmitted over the serial link. In the serializer, the 24-bit RGB pixel video stream is compressed (i.e., encoded) prior to transmission across the serial link. This results in a 3:1 reduction in serial link video bandwidth usage. In the deserializer, the video stream received from the serial link is decompressed (i.e., decoded) to reconstruct a visually-lossless video. The reduced bandwidth consumption of DSC-compressed video streams allows for higher resolutions to be transmitted over the GMSL2 link. With DSC enabled, pixel clocks up to 600MHz are supported and full 4K60 video (3840x2160x60) can be transmitted.

Refer to the VESA DSC v1.1 specification for more information on the details of the compression algorithm and image quality.

14.2 Operation

14.2.1 GMSL2 DSC Implementation Details

GMSL2 devices implement DSC compression with a fixed 3:1 compression ratio. Each video frame is split in half along the center of the horizontal active period, forming two parallel DSC slices (compliant with VESA DSC v1.1 specifications). This horizontal division halves the PCLK through each DSC encoder, enabling support for PCLKs up to 600MHz. Additionally, the symmetric split enables the use of side-by-side dual-view without compression artifacts carrying over the border between left and right halves of the image (see the *Dual-View* section for more details). Compression is only available on serializer and deserializer GMSL2 PHY A (SIOAP/SIOAN pins); it is not available on GMSL2 PHY B nor in dual link mode (see GMSL2 Link Configurations for details). The maximum horizontal line width supported is 5760 pixels; the maximum vertical image size is limited only by GMSL2 bandwidth availability (see GMSL2 Link System Bandwidth).

The DSC encoder requires that the sync polarity must be positive for HS and VS. If needed, the video crossbar in the serializer can be used to invert the polarity. In the deserializer, the sync polarity is set in the eDP/DP video configuration registers and is independent of the setting(s) in the serializer. The default setting is positive polarity for both HS and VS signals.

Line timing of a DSC compressed video is the same as an uncompressed video line, however, the video bandwidth over the GMSL2 link is reduced by a factor of three.

DSC compression does not require a frame buffer; line buffers are used in devices which support compression. The maximum latency when using compression is two video lines for any video less than 1920 pixels wide and three video lines for video streams wider than 1920 pixels.

ADI Confidential for New Revision under NDA
Analog Devices Confidential, Subject to ADI Non-Disclosure Agreement (NDA)
Maxim Integrated Confidential,
Provided to CMSL2_Customers Under NDA

14.2.2 Effects of Bit Errors

The DSC compression algorithm is susceptible to bit errors on the serial link: a single bit error can corrupt a full DSC macroblock and cause video disruption(s) detectible by the human eye. GMSL2 devices that support DSC compression implement forward error correction (FEC) to detect and correct potential bit errors. FEC significantly reduces the potential for bit errors when DSC compression is used. Refer to the Forward Error Correction (FEC) section for additional information.

14.2.3 Safety Mechanisms

Two safety mechanisms, video frame CRC and DSC encoder self-test, are available for GMSL2 devices to ensure video data integrity across the serial link while using DSC compression.

14.2.3.1 Video Frame CRC

When enabled in the serializer, video frame cyclic redundancy checks (CRC) are computed both at the video received on the HDMI input and at the output of the video pipe at the same time. These two CRCs are compared to check for video corruption. A mismatch in the CRC computation is reported to a register and can also be routed to the ERRB pin. Reference the CRC Error Detection and ARQ Error Correction section for additional information.

14.2.3.2 DSC Encoder Self-Test

The DSC compression block has a built-in self-test (BIST) safety measure that can be enabled by a sequence of register writes. First, the BIST is run on the 16 memory blocks used by the DSC encoder/decoder. This is then followed by a compressed frame CRC check in which all incoming pixels are replaced with a constant value of 0x5AA5C3. At the output, the pixels are checked to ensure that there are no errors in the DSC encode operation. This self-test takes four frames to complete.

14.3 Configuration

DSC configuration involves programming both the serializer and deserializer. The DSC blocks must be enabled in both devices. Additionally, the picture parameter set (PPS) must be programmed in accordance with the DSC specifications. The PPS parameters are contained in a block of approximately 100 registers. For simplified setup, 15 register-selectable “video profiles” of standard video resolutions are available. These video profiles are available in Table 31. For all other video resolutions, a configuration tool is available to create the PPS parameter register writes. For all video resolutions, the PPS parameters for the serializer and deserializer must match to ensure proper operation.

ADI Confidential for New Vision under NDA
Analog Confidential Distribution Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

14.3.1 Video Profiles

Table 31 Abbreviations:

- Horizontal total pixels = Htot
- Horizontal active pixels = Ha
- Horizontal front porch = Hfp
- Horizontal sync width = Hsw
- Horizontal back porch = Hbp
- Vertical total pixels = Vtot
- Vertical active pixels = Va
- Vertical front porch = Vfp
- Vertical sync width = Vsw
- Vertical back porch = Vbp

Table 31. Video Timings for Pre-defined Video Profiles

Profile #	Pclk (MHz)	FPS	Hres	Vres	Htot	Vtot	Hfp	Hsw	Hbp	Vfp	Vsw	Vbp
0	265.82	60	4800	900	4890	906	30	30	30	2	2	2
1	135.36	60	2400	900	2490	906	30	30	30	2	2	2
2	**RESERVED**											
3	104.56	30	1920	1728	2010	1734	30	30	30	2	2	2
4	201.53	60	1856	1720	1946	1726	30	30	30	2	2	2
5	178.32	60	1624	1728	1714	1734	30	30	30	2	2	2
6	594.00	60	3840	2160	4400	2250	176	88	296	8	10	72
7	148.50	60	1920	1080	2200	1125	88	44	148	4	5	36
8	381.19	60	5760	1080	5850	1086	30	30	30	2	2	2
9	594.00	60	4096	2160	4400	2250	88	88	128	8	10	72
10	348.58	60	2560	1600	3504	1658	192	280	472	3	6	49
11	74.25	60	1280	720	1650	750	110	40	220	5	5	20
12	221.95	30	2640	2704	2730	2710	30	30	30	2	2	2
13	205.79	60	1888	1728	1978	1734	30	30	30	2	2	2
14	330.64	60	3088	1728	3178	1734	30	30	30	2	2	2
15	135.02	60	1624	1000	2168	1038	296	168	80	3	10	25

Note: Do not use video Profile 2. It is reserved.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customer Under NDA

14.3.2 "Video Profile" Resolution

To enable DSC with an available video profile, the profile must be loaded in both serializer and deserializer. The DSC blocks in both devices must be configured before video is sent but can be configured in any order. Serializer configuration is presented in Table 32; deserializer configuration is presented in Table 33.

Table 32. Video Profile: Serializer Configuration

Description	Register Name	Register Write
Ensure VS and HS polarity is positive	HS_INV and VS_INV	Enable polarity invert if HS and/or VS has negative polarity
Put DSC block in reset	COM_MAIN_CONF_B1	dsce_swrst = 1
Select desired video profile	COM_MAIN_CONF_B1	vid_sw_prof_sel[3:0] = desired profile in hex
Load selected video profile	COM_MAIN_PROF_B0	cmd_ld_prof_def = 1
Enable DSC	DSC_CTRL	dsc_en = 1
Release DSC block from reset	COM_MAIN_CONF_B1	dsce_swrst = 0 (do not overwrite video profile)

Table 33. Video Profile: Deserializer Configuration

Description	Register Name	Register Write
Select desired video profile	DSCD_CTRL_B1	prof_sel[3:0] = desired profile in hex
Load selected video profile	DSCD_CTRL_B1	prof_wr = 1
Enable DSC and reset to clear out any data from initialization	DSCD_CTRL_B0	dscd_en = 1 and dscd_rst_sc = 1

14.3.2.1 Example Script

The following example is for an HDMI serializer at address 0x80 and eDP/DP deserializer at address 0x90, loading video profile 7 for VESA standard 1080p video timing.

```
# Serializer setup
# Put DSC block in reset
0x80,0x4401,0x80
# Select Profile 7
0x80,0x4401,0x87
# Load profile
0x80,0x4404,0x01
# Enable DSC
0x80,0x025F,0x01
# Release DSC block from reset
0x80,0x4401,0x07
###############
# Deserializer Setup
# Select Profile 7
0x90,0x4501,0x07
# Load profile
0x90,0x4501,0x87
# Enable DSC
0x90,0x4500,0x05
```

ADI Confidential for Newvision under NDA
Maxim Confidential
Proprietary to GMSL2_ Customers Under NDA
Maxim Integrated Company Confidential,

14.3.3 Custom Video Resolution

If the video resolution does not match any of the pre-defined video profiles, it is recommended to use the DSC calculator tool to generate the register script needed for DSC operation.

14.3.4 Video Frame CRC Safety Mechanism

In the GMSL2 serializer, enable a video frame CRC calculation at the HDMI Rx and the video pipe output by setting register hdmi_crc_err_oen = 1. Check the result (i.e., if video frame CRC calculation at HDMI RX and video pipe output match) by reading register hdmi_crc_err_flag. This bit will flag an error if the CRC calculations do not match.

CRC mechanisms are available for the video transport over the serial link. Refer to the CRC Error Detection and ARQ Error Correction section for additional options.

Note: The video frame CRC safety mechanism is driven by the serializer and does not require configuration of the deserializer.

14.3.5 DSC Encoder Self-Test for Video Profile Resolutions

If a video profile is being used, use the following configuration to run the DSC self-test on device power-up (Table 34).

Note: The DSC encoder self-test safety mechanism is driven by the serializer and does not require configuration of the deserializer.

Table 34. Serializer Configuration for DSC Encoder Self-Test (for Video Profile Resolutions)

Description	Register Name	Register Write
Put DSC block in reset	COM_MAIN_CONF_B1	dsce_swrst = 1
Select desired video profile	COM_MAIN_CONF_B1	vid_sw_prof_sel[3:0] = desired profile in hex
Load selected video profile	COM_MAIN_PROF_B0	cmd_ld_prof_def = 1
Enable DSC and the DSC encoder self-test	DSC_CTRL	dsc_en = 1 dsce_selftest = 1
Release DSC block from reset	COM_MAIN_CONF_B1	dsce_swrst = 0 (do not overwrite video profile)
Wait (Poll) for self-test to finish	DSC_STATUS	Test finished when dsce_selftest_done = 1
Check the self-test error status. First check the result of the BIST	DSC_STATUS	dsce_selftest_mb_fail
Check the self-test error status. Second check the result of the frame CRC test	DSC_CTRL	dsce_selftest_crc_err
Check the self-test error status. Finally, check the overall latched error flag	DSC_STATUS	dsce_selftest_err_flag

ADI Confidential for New vision under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

14.3.6 DSC Encoder Self-Test for non-Video Profile Resolutions

The pre-stored DSC frame CRC registers are not writable, so the DSC self-test procedure for a non-profile video resolution requires that the VTG re-time the video to 4K60. Once the video is re-timed to 4K60 (matching DSC video profile 6), the pre-stored DSC frame CRC can be utilized. This configuration sequence is given in *Table 35*.

Note: The DSC encoder self-test safety mechanism is driven by the serializer and does not require configuration of the deserializer.

Table 35. Serializer Configuration for DSC Encoder Self-Test (for non-Video Profile Resolutions)

Description	Register Name	Register Write
Disable video in serializer video pipe	DV2	VID_EN_X = 0
Disable video line CRC	VID_TX__X VIDEO_TX0	LINE_CRC_EN = 0
Configure VTG to standard 4K60 timing		
VTG: Use HDMI input PCLK as clock source	VTX1	0x09
VTG: Program VS_DLY	VTX2	0x00
VTG: Program VS_DLY	VTX3	0x02
VTG: Program VS_DLY	VTX4	0x58
VTG: VS high time in terms of superframe pixels	VTX5	0x00
VTG: VS high time in terms of superframe pixels	VTX6	0x00
VTG: VS high time in terms of superframe pixels	VTX7	0x00
VTG: VS low time in terms of superframe pixels	VTX8	0x00
VTG: VS low time in terms of superframe pixels	VTX9	0x00
VTG: VS low time in terms of superframe pixels	VTX10	0x00
VTG: Time between VS edge of superframe and first VTX HS generation. Equal to VS_DLY to keep VS and HS aligned.	VTX11	0x00
VTG: Time between VS edge of superframe and first VTX HS generation. Equal to VS_DLY to keep VS and HS aligned.	VTX12	0x00
VTG: Time between VS edge of superframe and first VTX HS generation. Equal to VS_DLY to keep VS and HS aligned.	VTX13	0x10
VTG: Double HS high time in terms of superframe pixels	VTX14	0x00
VTG: Double HS high time in terms of superframe pixels	VTX15	0x00
VTG: Double HS low time in terms of superframe pixels	VTX16	0x00
VTG: Double HS low time in terms of superframe pixels	VTX17	0x00
VTG: Half the number of HS pulses in the superframe	VTX18	0x00
VTG: Half the number of HS pulses in the superframe	VTX19	0x00
VTG: Time between VS rising edge of superframe and first VTG DE generation.	VTX20	0x00
VTG: Time between VS rising edge of superframe and first VTG DE generation.	VTX21	0x00
VTG: Time between VS rising edge of superframe and first VTG DE generation	VTX22	0x00
VTG: Double DE high time in a line terms of superframe pixels	VTX23	0x00
VTG: Double DE high time in a line terms of superframe pixels	VTX24	0x00

Maxim Confidential for Newvision under NDA
Provided to GMSL2 Customers Under NDA

VTG: Double DE low time in a line terms of superframe pixels	VTX25	0x00
VTG: Double DE low time in a line terms of superframe pixels	VTX26	0x00
VTG: Half the number of lines in the superframe	VTX27	0x00
VTG: Half the number of lines in the superframe	VTX28	0x00
VTG: Enable HS, VS, and DE generation and use free running trigger mode	VTX0	VTG_MODE[1:0] = 11 GEN_DE = 1 GEN_HS = 1 GEN_VS = 1
Put DSC block in reset	COM_MAIN_CONF_B1	dsce_swrst = 1
Configure DSC for profile 6	DSCD_CTRL_B1	prof_sel[3:0] = 0110
Load profile 6	DSCD_CTRL_B1	prof_wr = 1
Delay 2 frames		
Enable DSC and the self-test	DSCD_CTRL_B0	dsc_en = 1 dsce_selftest = 1
Release DSC block from reset	COM_MAIN_CONF_B1	dsce_swrst = 0 (do not overwrite video profile)
Wait (Poll) for self-test to finish	DSC_STATUS	Test finished when dsce_selftest_done = 1
Check the self-test error status. First check the result of the BIST	DSC_STATUS	dsce_selftest_mb_fail
Check the self-test error status. Second check the result of the frame CRC test	DSC_CTRL	dsce_selftest_crc_err
Check the self-test error status. Finally check the overall latched error flag	DSC_STATUS	dsce_selftest_err_flag
Disable VTG	VTX0	GEN_DE = 0 GEN_HS = 0 GEN_VS = 0
Enable video line CRC	VID_TX__X VIDEO_TX0	LINE_CRC_EN = 1
Enable video in serializer video pipe	DV2	VID_EN_X = 1
Configure DSC for custom resolution with registers provided in the DSC configuration tool		

ADI Confidential for Newvision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

14.4 Status and Debug Registers

The following status registers are available in the serializers and deserializers which support DSC compression or decompression. The status registers may indicate an error state if compression is not yet enabled or video is not present. After the system is fully initialized and video is enabled, the status registers should be cleared and re-checked. On serializers, compression errors can be enabled to the ERRB pin, on deserializer, de-compression errors can be enabled to the ERRB pin.

14.4.1 Serializer DSC Status Registers

DSC encoder overflow and underflow status registers are available to read via the main control channel (see Table 36). These errors can be individually enabled (Table 37) to be output to the overall DSC error status register (all error outputs are enabled by default). This can be reported to the ERRB pin (along with the DSC self-test status) by setting dsce_err_oen = 1.

Table 36. DSC Encoder Status Registers

Register	Bitfield	Description
HS0_ENC_INT_STAT_B0	enc_underflow_context_0	DSC Slice 0 (left half of frame) encoder underflow
HS0_ENC_INT_STAT_B0	rc_model_buffer_fullness_overflow_context_0	DSC Slice 0 (left half of frame) buffer overflow
HS1_ENC_INT_STAT_B0	enc_underflow_context_0	DSC Slice 1 (right half of frame) encoder underflow
HS1_ENC_INT_STAT_B0	rc_model_buffer_fullness_overflow_context_0	DSC Slice 1 (right half of frame) buffer overflow

Table 37. DSC Encoder Error Output Registers

Register	Bitfield	Description
HS0_ENC_INT_MASK_B0	enc_underflow_context_0	Disable error reporting for DSC Slice 0 (left half of frame) encoder underflow
HS0_ENC_INT_MASK_B0	rc_model_buffer_fullness_overflow_context_0	Disable error reporting for DSC Slice 0 (left half of frame) buffer overflow
HS1_ENC_INT_MASK_B0	enc_underflow_context_0	Disable error reporting for DSC Slice 1 (right half of frame) encoder underflow
HS1_ENC_INT_MASK_B0	rc_model_buffer_fullness_overflow_context_0	Disable error reporting for DSC Slice 1 (right half of frame) buffer overflow

ADI Confidential for Newvision Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

Table 38. DSC Self-Test Status Registers

Register	Bitfield	Description
DSC_STATUS	dsce_selftest_done	Two-step self-test completed
DSC_STATUS	dsce_selftest_err	Self-test operation had error
DSC_STATUS	dsce_selftest_mb_done	MBIST step of self-test completed
DSC_STATUS	dsce_selftest_mb_fail	MBIST step of self-test had error
DSC_STATUS	dsce_selftest_err_flag	Latched err flag used for ERRB contribution; cleared on read
DSC_STATUS	hdmi_crc_err_flag	Latched err flag used for ERRB contribution; cleared on read
DSC_CTRL	dsce_selftest_crc_err	Frame CRC step of self-test had error; cleared on read of 0x25E

Table 39. DSC Encoder ERRB Output

Register	Bitfield	Description
DSC_CTRL	dsce_err_oen	Enable reporting of DSC errors to the ERRB pin

14.4.2 Deserializer DSC Status Registers

DSC decoder overflow and underflow status registers are available to read via the main control channel (see Table 40). These errors can be individually enabled (Table 41) to be output to the overall DSC error status register (all error outputs are disabled by default). This can be reported to the ERRB pin by setting dscd_err_oen = 1.

Note: To check DSC error status, the bit should be written to a 1 to clear and then can be read back for status

Table 40. DSC Decoder Status Registers

Register	Bitfield	Description
DSCD_INT_ST_B0	vb_ovfl_ist	Video buffer overflow
DSCD_INT_ST_B0	pb_ovfl_ist	Pixel buffer overflow
DSCD_INT_ST_B1	ib0_ovfl_ist	Decoder core input buffer slice 0 (left half of frame) overflow
DSCD_INT_ST_B1	ib0_undfl_ist	Decoder core input buffer slice 0 (left half of frame) overflow
DSCD_INT_ST_B1	tc_size0_err_ist	Decoder core slice 0 (left half of frame) transport chunk size error
DSCD_INT_ST_B1	rcb0_ovfl_ist	Decoder core slice 0 (left half of frame) rate control model buffer fullness overflow
DSCD_INT_ST_B1	ib1_ovfl_ist	Decoder core input buffer slice 1 (right half of frame) overflow
DSCD_INT_ST_B1	ib1_undfl_ist	Decoder core input buffer slice 1 (right half of frame) underflow
DSCD_INT_ST_B1	tc_size1_err_ist	Decoder core slice 1 (right half of frame) transport chunk size error

ADI Confidential for Newvision Under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA

DSCD_INT_ST_B1	rcb1_ovfl_ist	Decoder core slice 1 (right half of frame) rate control model buffer fullness overflow

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Table 41. DSC Decoder Error Output Registers

Register	Bitfield	Description
DSCD_INT_EN_B0	vb_ovfl_ien	Video buffer overflow error output enable
DSCD_INT_EN_B0	pb_ovfl_ien	Pixel buffer overflow error output enable
DSCD_INT_EN_B1	ib0_ovfl_ien	Decoder core input buffer slice 0 (left half of frame) overflow error output enable
DSCD_INT_EN_B1	ib0_undfl_ien	Decoder core input buffer slice 0 (left half of frame) overflow error output enable
DSCD_INT_EN_B1	tc_size0_err_ien	Decoder core slice 0 (left half of frame) transport chunk size error output enable
DSCD_INT_EN_B1	rcb0_ovfl_ien	Decoder core slice 0 (left half of frame) rate control model buffer fullness overflow error output enable
DSCD_INT_EN_B1	ib1_ovfl_ien	Decoder core input buffer slice 1 (right half of frame) overflow error output enable
DSCD_INT_EN_B1	ib1_undfl_ien	Decoder core input buffer slice 1 (right half of frame) underflow error output enable
DSCD_INT_EN_B1	tc_size1_err_ien	Decoder core slice 1 (right half of frame) transport chunk size error output enable
DSCD_INT_EN_B1	rcb1_ovfl_ien	Decoder core slice 1 (right half of frame) rate control model buffer fullness overflow error output enable

Table 42. DSC Decoder ERRB Output

Register	Bitfield	Description
INTR6	dscd_err_oen	Enable reporting of DSC errors to the ERRB pin

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

14.5 Feature Availability by Device Family

DSC compression is available in select HDMI serializers as well as select eDP/DP and CSI-2 deserializers. See the table below for part numbers which support DSC compression (Table 43).

Table 43. DSC Support by Part Number

Device Family	DSC Compression	DSC Decompression	DSC Pass-through
HDMI Serializers			
Advanced HDMI Serializers	MAX96763 MAX96765 MAX96759 MAX96761		
DSI Serializers			
Advanced DSI Serializers			
CSI-2 Serializers			
Advanced CSI-2 Serializers			
CSI-2 Camera Deserializers		MAX96784 MAX9296C	
CSI-2 Display Deserializers			
OLDI Deserializers			
eDP/DP Deserializers		MAX96772 MAX96774 MAX96776 MAX96778	
CSI-2 Quad Deserializers			

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Subject to Maxim Integrated Company Confidential, Provided to GMSL2_Customers under NDA

15 Forward Error Correction (FEC)

15.1 Overview

Forward Error Correction (FEC) is used in GMSL2 devices to detect and correct bit errors occurring during the transmission of compressed video on the serial link. The FEC implementation in GMSL2 devices corrects both single and burst errors and results in an improved bit-error rate (BER). An additional CRC mechanism is used to indicate the presence of uncorrectable bit errors. FEC consumes an additional fixed 6.7% bandwidth overhead when enabled; however, it provides error correction without requiring reverse channel communication or incurring video stream delays.

FEC is primarily used to enable robust transmission of compressed video over a GMSL2 link. On an uncompressed video stream, a single bit error on a video packet will corrupt only one pixel and is visually undetectable (*Figure 35*). However, on a compressed video stream, a single bit error can possibly corrupt the entirety of the DSC macroblock, typically comprising thousands of pixels, and can usually be detected visually (*Figure 36*).

To test the visual impact of bit errors on a compressed video stream, 1000 compressed 4K frames, each with a single bit error, were transmitted over a GMSL2 link with FEC disabled. The average observed macroblock error was 484x48 pixels (0.28% of the image) and was visibly detectable approximately 60% of the time.

Since a single bit error can cause visual corruption of a compressed video stream, FEC is required for all GMSL links with compressed video to minimize the BER so that the probability of bit errors becomes negligible.

Note: Refer to the *Display Stream Compression (DSC)* section for additional information regarding DSC macroblocks.



Figure 35. Uncompressed Video Frame with 91,480 Random Bit Errors



Figure 36. Compressed Video Frame with 1 Bit Error

The above images contrast the visual impact of bit errors on uncompressed and compressed video frames. In *Figure 35*, the 91,480 random bit errors are undetectable. However, in *Figure 36*, a single bit error on a video packet has corrupted an entire DSC macroblock and resulted in visually detectable damage to the video frame.

ADI Confidential for New Version under NDA
Maxim Integrated GMSL Distributed Under NDA
Provided to GMSL Customers Under NDA
Company Confidential

15.2 Operation

15.2.1 Architecture

In the serializer, data is grouped into codeword blocks of 2560 bits and converted to GMSL data. After conversion, the GMSL data is encoded by the FEC block then processed by the 9b10b encoder/scrambler. The GMSL data is transmitted from the serializer's GMSL PHY over the serial link to the deserializer's GMSL PHY. In the deserializer, the GMSL data is first processed by the 9b10b decoder and de-scrambler. This block detects decode and idle errors. Bit errors are detected and corrected when GMSL data passes through the FEC decoder. The count of errors corrected by the FEC decoder block are reported to fec_bit_errors_corrected. The GMSL data then moves to the FEC CRC block. An 18-bit CRC is calculated per codeword block. If the received CRC matches the calculated CRC, the block is deemed correct; if the CRC values do not match, it is deemed an uncorrectable codeword block and reported to fec_uncorrectable_blks. The block diagram below (Figure 37) illustrates the data path through the serial link system and highlights error detection, correction, and reporting behavior.

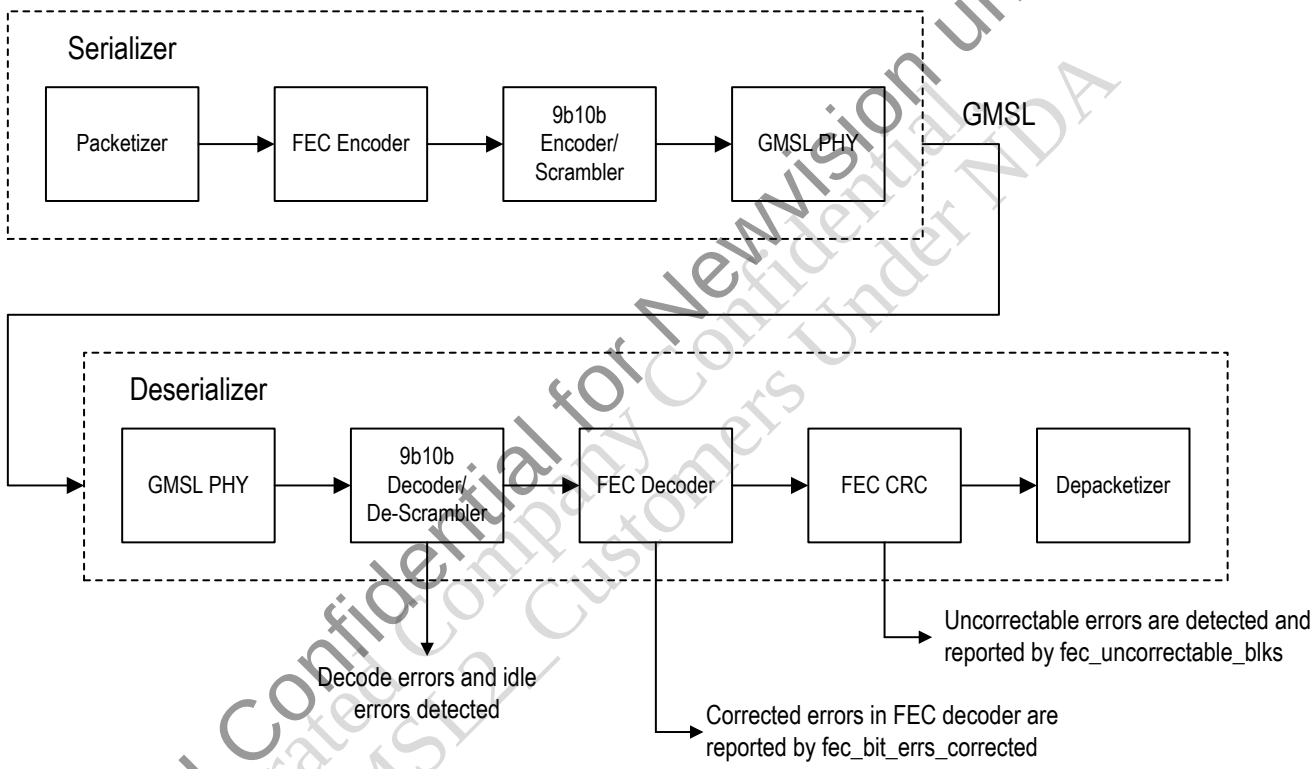


Figure 37. Architecture of FEC Implementation in GMSL Devices

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

15.2.2 BER Reduction Performance

FEC is designed to lower the BER on GMSL links containing compressed video streams. The FEC block operates with an error correction scheme based on Reed-Solomon error-correction codes. This coding scheme improves the BER in high-error systems (e.g., wireless transmission), and it is used to substantially improve the BER in GMSL systems and effectively mitigate the incidence of uncorrectable bit errors on compressed video streams. Without FEC enabled, compliant GMSL2 links operate with a BER range of approximately 10^{-12} to 10^{-16}; with FEC enabled, the GMSL2 links operate with an improved BER range of approximately 10^{-40} to 10^{-60}. To illustrate, a link operating with a BER of 10^{-15} is improved to a BER of 10^{-55} with FEC enabled, which corresponds to a single uncorrectable bit error every 1.6 x 10^{38} years.

15.2.3 Bandwidth Overhead and System Impacts

The overall FEC bandwidth overhead is the combination of the Reed-Solomon correction overhead and the 18-bit CRC correction overhead. The code rate of Reed-Solomon [N/K/S] = [127/121/7] plus an additional symbol to pack parity bits and the CRC results in a total bandwidth overhead of 128/120 = 6.677%.

The link overhead added when FEC is enabled must be considered when calculating GMSL bandwidth consumption. Additionally, FEC adds a small increase in the link latency of the video and control channel data that is proportional to the forward channel serial link rate. For 6Gbps link rate, the additional latency due to FEC encoding and decoding is 720ns; for 3Gbps link rate, the latency is 1440ns. I2C clock stretching, GPIO delay compensation, etc. can be used to compensate for this latency. See the I2C/UART and Interface-Specific Bandwidth Calculations sections for details.

15.2.4 Resynchronization

If a receiver loses and subsequently regains synchronization, FEC returns to normal operation without freezing. Therefore, FEC auto-recovers if the GMSL link lock is lost and recovered.

15.2.5 Power-up Configuration

The FEC block is disabled by default on all GMSL2 devices. It is enabled using the procedure in the Configuration section (below).

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

15.3 Configuration

FEC only requires a single register write per device to be enabled; additional configuration may be needed to change the ERRB reporting depending on the requirements of the application.

Ensure that FEC is either enabled or disabled on both serializer and deserializer. The deserializer FEC should be enabled before or simultaneous to the serializer. Note that valid operation cannot be guaranteed until both devices have been properly configured using the procedures presented below.

Once both devices have been configured, FEC will auto-recover if the link is lost and re-established or either device is reset (and re-initialized).

15.3.1 Enabling FEC in a Single Microcontroller System

The procedure below applies if all configuration of the serializer and deserializer is performed on one side of the link.

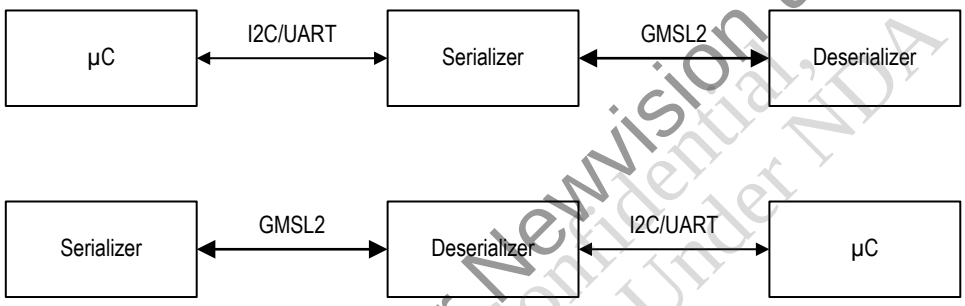


Figure 38. Block diagram of a Single Microcontroller System

In a single microcontroller system, the microcontroller can be on either side of the link (Figure 38). The remote device is configured over the serial link via the control channel.

15.3.1.1 Single Microcontroller System Configuration Procedure

1. Wait for link LOCK (ensure bit LOCKED = 1 on serializer or deserializer)
2. Enable FEC in the deserializer (set fec_en = 1)
3. Enable FEC in the serializer (set TX_FEC_EN = 1)

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Company Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

15.3.2 Enabling FEC in a Dual Microcontroller System

The below procedure describes how to enable FEC in a system that has a microcontroller on both sides of the link and has the control channel disabled (Figure 39). The devices can be programmed in any order provided that the link is held in reset and link lock is not established before configuration is completed.

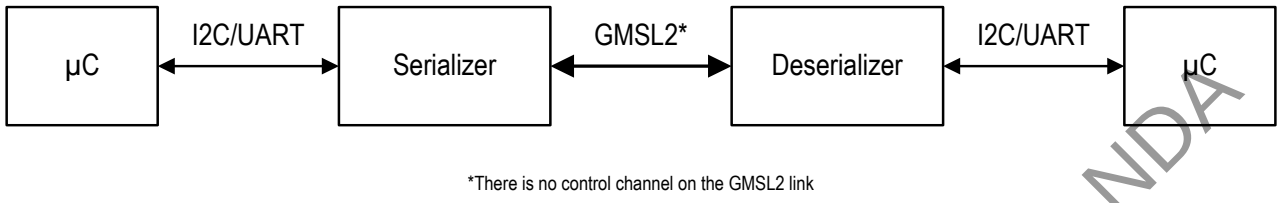


Figure 39. Block Diagram of a Dual Microcontroller System

15.3.2.1 Deserializer Programming Procedure

1. Write RESET_LINK = 1 (this prevents premature link lock).
2. Write other initialization registers (configuration of video, control channels, etc.).
3. Enable FEC (set fec_en = 1).
4. Clear all ERRs.
5. Write RESET_LINK = 0 (this allows link to lock when both devices are ready).

15.3.2.2 Serializer Programming Procedure

1. Write RESET_LINK = 1 (this prevents premature link lock).
2. Write other initialization registers (configuration of video, control channels, etc.).
3. Enable FEC (set TX_FEC_EN = 1).
4. Write RESET_LINK = 0 (this allows link to lock when both devices are ready).

ADI Confidential for review under NDA
Analog Devices Confidential Distribution only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

15.4 Status and Debug Registers

FEC enables reporting of uncorrected link errors to the ERRB pin, as well as providing status registers to determine the exact link BER rate and post-correction BER rate. Reference the GMSL2 Error Reporting (ERRB Pin) section for additional information.

15.4.1 Error Reporting to the ERRB pin

When the number of uncorrectable FEC errors is greater than FEC_ERR_THR (set to 1 by default), the FEC_RX_ERR_FLAG is set to 1. If FEC_RX_ERR_OEN = 1, the FEC_RX_ERR_FLAG error sets the ERRB pin high.

When FEC_RX_ERR_FLAG is read, it clears to 0. To reset error reporting so that additional FEC errors will be detected, the FEC statistics counters must also be reset by setting fec_clr_stats = 1 (this bit self-clears after writing).

15.4.1.1 Example System Pseudocode

```
if Deserializer ERRB pin goes low and triggers hardware interrupt:
      # Read error registers to determine what error is active
      Read registers 0x001B, 0x001D, and 0x001F
      if FEC_RX_ERR_FLAG = 1
            react to error as defined by system engineer
            write fec_clr_stats = 1 to clear error counters
            read FEC_RX_ERR_FLAG to clear error
      end
end
```

15.4.2 Statistics Registers

The FEC block has status registers that indicate the number of uncorrected errors, number of corrected errors, and total number of codeword blocks processed. These status registers can be cleared at any time by writing fec_clr_stats = 1.

1. fec_blks_processed [31:0] – Blocks processed: number of codeword blocks received divided by 32768. For example, a value of 10 indicates that 327,680 blocks have been received. The scaling down by 32768 is implemented inside chip to avoid saturating the counter too soon. Write to clear or clear by writing fec_clk_stats = 1.
2. fec_uncorrectable_blks [31:0] – Uncorrectable errors: Number of codeword blocks that were uncorrectable by FEC (as detected by CRC). Write to clear or cleared by writing fec_clk_stats = 1.
3. fec_bit_errs_corrected [31:0] – Corrected bit errors: Number of bit errors corrected in the FEC block. Note that this does not map directly to the number of bit errors on the serial link (see *Link BER*). Write to clear or cleared by writing fec_clr_stats = 1.

Analog Confidential Distribution Only
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA
Analog Confidential for New Version under NDA

15.4.3 BER Calculations

For any system, the measured BER is given by the equation:

$$BER = \frac{Bit\ Errors}{Total\ Bits}$$

However, since the observation period for bit errors can be prohibitively long in systems with a low BER, an estimated BER can be determined for a given observation period of no bit errors using a Poisson distribution. In this case, the confidence level that a system is operating at or better than a given BER after observing no bit errors over a period of N_{bits} is given by:

$$Confidence\ Level\ (CL) = 1 - e^{BER * N_{bits}}$$

Rearranging and solving for a 95% confidence level:

$$BER_{estimated} = \frac{-\ln(0.05)}{N_{bits}}$$

These equations can be combined with the status registers in the FEC block to get the measured and estimated BER for a system.

15.4.3.1 Link BER

The Link BER (pre-FEC bit corrections) can be a good indicator of the health of the link, but it is not the BER seen by the system. To calculate the link BER, the number of bit errors and total bits in an observation period must be known.

Due to GMSL2 9b10b encoding, bit errors on the serial link do not map directly to bit errors seen by the FEC decoder block. An error on the link has a probability to cause between one and nine errors within the FEC decoder block. On average, a single bit error corrupts approximately 4.1 bits in the FEC block, so a scaling factor of 4.1 is applied to the BER approximation equations.

The approximate link BER is given by:

$$BER = \frac{fec_bit_errs_corrected}{4.1 * 2560 * 32768 * blks_processed}$$

If no bit errors are observed (fec_bit_errors_corrected = 0), using the Poisson distribution equation from above:

$$BER_{estimated} = \frac{-\ln(0.05)}{4.1 * 2560 * 32768 * blks_processed}$$

ADI Confidential for Review under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Confidential,
Provided to GMSL2_Customers Under NDA

The approximate BER after the FEC block corrects errors is given by:

$$BER_{FEC\ OUTPUT} = \frac{fec_uncorrectable_blks}{4.1 * 2560 * 32768 * blks_processed}$$

There is a tool available in the GUI which provides the BER based on these equations.

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

15.5 Feature Availability by Device Family

FEC is available on the same devices that support DSC compression: select HDMI serializers as well as select eDP/DP and CSI-2 deserializers. See *Table 44* for part numbers that support FEC.

Table 44. FEC Support by Part Number

Device Family	FEC Encoder	FEC Decoder
HDMI Serializers		
Advanced HDMI Serializers	MAX96763 MAX96765 MAX96759 MAX96761	
DSI Serializers		
Advanced DSI Serializers		
CSI-2 Serializers		
Advanced CSI-2 Serializers		
CSI-2 Camera Deserializers		MAX96784
CSI-2 Display Deserializers		
OLDI Deserializers		
eDP/DP Deserializers		MAX96772 MAX96774 MAX96776 MAX96778
CSI-2 Quad Deserializers		

ADI Confidential for New Revision under NDA
Analog Devices Confidential, Distribution Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

16 HDCP

Ful section pending.

16.1 GMSL HDCP 2.3 and loss of link lock/recovery

The GMSL link HDCP 2.3 requires periodic communication across the GMSL link to enable authentication, integrity checks, and general diagnostics. There is a vulnerability in the associated communication channel that may cause long delays of up to 5 seconds when turning on the HDCP 2.3 function in the event that the GMSL link connection is lost while HDCP 2.3 is active. This includes any time period during which the HDCP 2.3 function is powered up and operational. All serializers with GMSL HDCP 2.3 are susceptible to this vulnerability.

16.1.1 Avoiding long HDCP 2.3 turn-on

To avoid the long HDCP 2.3 turn-on time, link resets must be avoided after HDCP 2.3 is enabled. If the link is reset or lock is lost after HDCP 2.3 is enabled, the HDCP 2.3 soft reset sequence described below should be executed following the link being re-established to ensure fast recovery.

There are currently two device families that support HDCP 2.3. The behavior of the two differs slightly. The specific details for each of these families is detailed below.

In the case of Advanced HDMI Serializer devices, the HDCP 2.3 function is more susceptible to the long turn-on following a link reset or disruption. In addition, HDCP 2.3 is powered-up by default, increasing the probability that HDCP 2.3 will be enabled when the link is disrupted. As a result, these devices are most prone to the long HDCP 2.3 turn-on behavior, and it is strongly recommended that the user carefully consider the initialization sequence and use of the soft reset sequence following link disruptions.

eDP Serializers that include HDCP 2.3 are less susceptible to the long HDCP 2.3 turn-on following a link disruption when HDCP 2.3 is active. In addition, the HDCP 2.3 function in these devices is powered off by default, further reducing the probability of the long HDCP turn-on. When using an eDP Serializer, the simplest workaround is to delay powering up the HDCP 2.3 until after the last link lock event.

16.1.1.1 HDCP 2.3 Soft Reset

If the HDCP 2.3 soft reset routine is used, it must be issued after the last GMSL link lock event and immediately prior to HDCP 2.3 being enabled. The sequence can also be used to reset the HDCP 2.3 following a link lock loss and recovery. The HDCP 2.3 soft reset sequence is:

1. Final link lock event
2. HDCP Reset Sequence
    1. Serializer CMD_RESET->1 (0x4100[7])
    2. Deserializer CMD_RESET->1 (0x1700[7])
    3. Deserializer CMD_RESET->0 (0x1700[7])
    4. Serializer CMD_RESET->0 (0x4100[7])
3. Enable HDCP

Analog Devices Confidential Distribute Only Under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA
ADI Confidential for Newvision under NDA

17 Watermarking

17.1 Overview

The watermarking (WM) feature, available on all GMSL2 devices, is designed to detect frozen frames caused by SoCs in safety-critical applications. GMSL2 parts can both watermark safety-relevant video streams with a time-varying watermark or detect a watermarked video stream. It is implemented in a system by enabling the watermark in the first GMSL2 device in the video chain and detecting the watermark in the final device in the system. A frozen frame failure is indicated if the watermark detector fails to see all the generated watermarks between the generator and detector in a frame-based processing system (note that frozen frames that occur before the watermark generator or after the watermark detector will not be detected). In this error condition, the watermark detector can generate an interrupt and/or blank the output video and restore the display to a safe state in less than 500ms.

The watermark feature incorporates the following capabilities:

- A highly redundant design that is robust to image processing
- Use of four unique watermarks ensures frozen frame failures are detected in a video processing path with up to a three-frame video buffer.
- The watermark is embedded in the LSBs of the video images and is removed by the detector; frame rate control is used to restore the image quality with visually undetectable processing.
- Video streams without watermarks are not modified by the detector.
- The watermark control registers are keyed to prevent inadvertent disabling in an active system.
- The watermark processing supports RGB888 datatype only.
- End-to-end test capability is enabled with error generation: the error condition is emulated by only generating two of four watermarks.
- The watermark detect signal can be output to a GPIO on eDP/DP and OLDI deserializers to provide an interrupt when watermark detection is lost.

Note: The watermark block is only on Pipe X (except for CSI-2 Quad Deserializers). See *Feature Availability by Device Family*.

The watermark feature is immune to the following image processing:

- Image resizing (50%–200%)
- Image rotation (±5 degrees)
- Obstructions of up to 25% of the image area
- Contrast adjustment
- Color compensation
- Image horizontal and vertical flipping

ADI Confidential for review under NDA
Analog Devices Inc. Distribution only to Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

17.2 Watermark Operation

An example application is illustrated in Figure 40. Here, a surround view system generates a human-viewable video stream that is routed through a non-ASIL processor to a console display. The serializer on the surround view system embeds the time-varying watermarks in the outbound video stream. The deserializer on the display interface continuously monitors for watermarked video streams. If a watermarked video stream is detected, the video stream is considered safety-relevant, and the detector confirms that the expected time-varying watermarks are present. If a watermarked video stream does not contain all of the expected watermarks, the watermark detector signals an error condition (e.g., the head unit is generating a static image) and the detector asserts an error interrupt and/or video stream blanking.

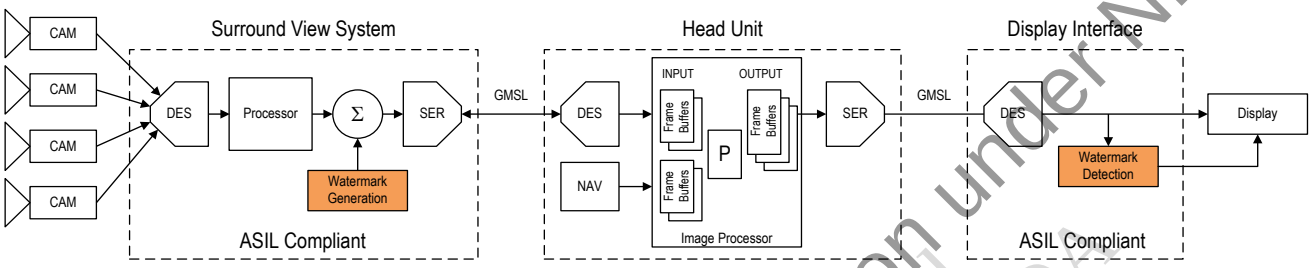


Figure 40. Surround View System Watermark Application

17.2.1 Watermark on Links with DSC Compression

A constant color (homogeneous) region is required for robust watermark detection on links with DSC compression enabled. The homogeneous area must be larger than 32 watermarks, calculated by:

$$rounddown\left(\frac{width}{320}\right) * height \geq 32 \text{ watermarks}$$

Note: Watermark is not generated in top line of image.

17.2.2 Watermark Register Configuration

GMSL2 serializer and deserializers can generate or detect watermarks. Register configurations are based on the watermark generator and watermark detector located in the following register blocks:

1. WM_0 – WM_7
2. WM0_0 – WM0_7
3. WM1_0 – WM1_7
4. WM_WREN_0 – WM_WREN_1

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

17.2.3 Modes of Operation

- Watermark Generator
- Watermark Generator (with Error Generation)
- Watermark Detector
- Watermark Detector with Watermark LOCK Output to GPIO2
- Watermark Detector with Watermark ERR Output to ERRB
- Watermark Detector with Automatic Blank on WM ERR
- Watermark Detector with WM DET Filtering

17.2.3.1 Watermark Generator

Unlock registers:	WM_WREN_L = 0xBA
	WM_WREN_H = 0xDC
Enable the generator:	WM_EN = 1'b1
Lock registers:	WM_WREN_L = 0x00
	WM_WREN_H = 0x00

17.2.3.2 Watermark Generator (with Error Generation)

Unlock registers:	WM_WREN_L = 0xBA
	WM_WREN_H = 0xDC
Enable the generator:	WM_EN = 1'b1
Enable ERR mode:	WM_MODE[2:0] = 3'b001
Lock registers:	WM_WREN_L = 0x00
	WM_WREN_H = 0x00

17.2.3.3 Watermark Detector

Unlock registers:	WM_WREN_L = 0xBA
	WM_WREN_H = 0xDC
Configure as Detector:	WM_DET[1:0] = 2'b01
Enable the Detector:	WM_EN = 1'b1
Set the threshold correctly:	WM_TH[6:0] = 0x0E
Lock registers:	WM_WREN_L = 0x00
	WM_WREN_H = 0x00

17.2.3.4 Watermark Detector with Watermark LOCK Output to GPIO2

17.2.3.4.1 eDP/DP Deserializers

Unlock registers:	WM_WREN_L = 0xBA
	WM_WREN_H = 0xDC
Configure as Detector:	WM_DET[1:0] = 2'b01
Enable the Detector:	WM_EN = 1'b1
Set the threshold correctly:	WM_TH[6:0] = 0x0E
Enable output to GPIO:	WM_DET_OUT_EN = 1'b1
Configure GPIO2 to output:	GPIO_OUT_DIS = 1'b0
Lock registers:	WM_WREN_L = 0x00
	WM_WREN_H = 0x00

ADI Confidential for Newvision under NDA
Maxim Integrated Proprietary Confidential, Provided to GMSL 2 Customers Under NDA

17.2.3.4.2 OLDI Deserializers

Unlock registers: WM_WREN_L = 0xBA
WM_WREN_H = 0xDC
Configure as Detector: WM_DET[1:0] = 2'b01
Enable the Detector: WM_EN = 1'b1
Set the threshold correctly: WM_TH[6:0] = 0x0E
Enable output to GPIO: WME_EN = 1'b1
Lock registers: WM_WREN_L = 0x00
WM_WREN_H = 0x00

17.2.3.5 Watermark Detector with Watermark ERR Output to ERRB

Unlock registers: WM_WREN_L = 0xBA
WM_WREN_H = 0xDC
Configure as Detector: WM_DET[1:0] = 2'b01
Enable the Detector: WM_EN = 1'b1
Set the threshold correctly: WM_TH[6:0] = 0x0E
Enable Interrupt to ERRB: WM_ERR_OEN =1'b1
Lock registers: WM_WREN_L = 0x00
WM_WREN_H = 0x00

17.2.3.6 Watermark Detector with Automatic Blank on WM ERR

Unlock registers: WM_WREN_L = 0xBA
WM_WREN_H = 0xDC
Configure as Detector: WM_DET[1:0] = 2'b01
Enable the Detector: WM_EN = 1'b1
Set the threshold correctly: WM_TH[6:0] = 0x0E
Enable Video blanking: WM_MASKMODE[1:0] = 2'b01
Lock registers: WM_WREN_L = 0x00
WM_WREN_H = 0x00

17.2.3.7 Watermark Detector with WM DET Filtering

Unlock registers: WM_WREN_L = 0xBA
WM_WREN_H = 0xDC
Configure as Detector: WM_DET[1:0] = 2'b01
Enable the Detector: WM_EN = 1'b1
Set the threshold correctly: WM_TH[6:0] = 0x0E
Enable WM timer: WMtimer[7:0] – set to desired timer length. See Filter the Watermark Error Condition below.
Lock registers: WM_WREN_L = 0x00
WM_WREN_H = 0x00

ADI Confidential for New vision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL Customers Under NDA

17.2.4 System Implementation

It is recommended that the watermark generator and detector configuration be completed during initial systems configuration before the video stream is enabled. It should not be dynamically enabled/disabled during use.

17.2.5 Watermark Status Registers and Outputs

Figure 41 illustrates the WM status output options. In the WM detection circuit, there are three live status registers: WM_ERROR, WM_ERR_FLAG, and WM_DETOUT. There is also the ability to output WM_ERR_FLAG to ERRB pin and WM_DETOUT to GPIO2 (on eDP/DP and OLDI deserializers).

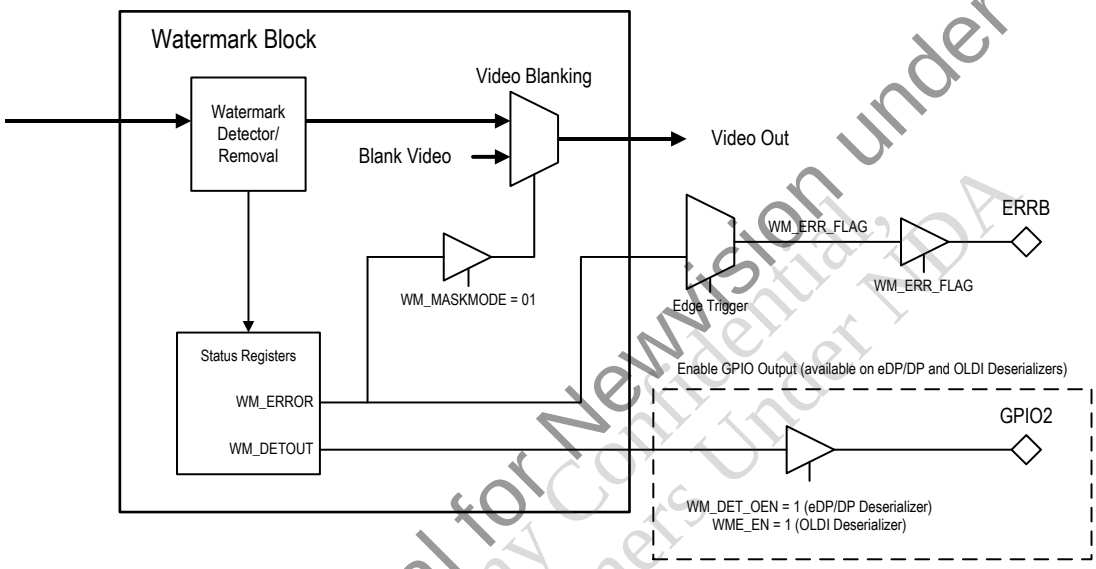


Figure 41. Watermark Status Outputs

ADI Confidential for New Version under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

17.3 Status and Debug Registers

The watermarking status can be monitored through the following status registers in the watermark detector:

WM_DETOUT – This bit indicates if a watermark is detected. It will be high if a watermark is detected in each frame and will be low if the watermark is not detected. This will stay high if there is a frozen frame condition because the watermark is still in the image. This bit status can be mapped out to a GPIO if desired (eDP/DP and OLDI deserializers only). Watermarks can only be detected if valid video is flowing.

WM_ERROR – This bit is a real-time flag indicating if a watermark error occurs. A watermark error is defined as a state in which all four unique watermarks are not detected in the previous eight frames. This differs from a frozen frame condition wherein the same watermark is repeated in every frame. The WM_ERROR register will go high if there is a watermark error and will immediately return low once the error condition no longer exists.

WM_ERR_FLAG – This error flag is triggered on the initial occurrence (edge) of the WM_ERROR bit. The status of this error flag is output to the ERRB pin (if WM_ERR_OEN = 1). This flag is cleared on read even if the WM error condition still exists. If WM_ERROR returns low and then is triggered on a second WM error event, this flag will be re-triggered.

17.3.1 ERRB Error Handling

The following is a suggested procedure for ERRB error handling:

- During nominal operation, use the ERRB output as an interrupt.
- When ERRB is active, check error registers (INTR3, INTR5, INTR7) to determine what error has occurred.
- If WM_ERROR occurs, react appropriately for frozen frame error handling. An option to automatically blank video to black is available (see *Watermark Detector with Automatic Blank on WM ERR* above).
- Monitor the WM_ERROR register until the frozen frame condition is fixed. The WM_ERROR register bit will return low when the frozen frame condition resolves.

Note: Reference the *GMSL2 Error Reporting (ERRB Pin)* section for additional information.

17.3.2 Status Bits Under Various Conditions

The state of registers for common conditions are listed below.

Default state

1. WM_DETOUT = 0
2. WM_ERROR = 0
3. WM_ERR_FLAG = 0
4. ERRB = 1

WM is enabled with no errors

- WM_DETOUT = 1
- WM_ERROR = 0
- WM_ERR_FLAG = 0
- ERRB = 1

ADI Confidential for review version under NDA
Analog Devices Confidential, Distribution Prohibited
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

WM is disabled with no errors

- WM_DETOUT = 0
- WM_ERROR = 0
- WM_ERR_FLAG = 0
- ERRB = 1

WM is enabled with lost video signal

- WM_DETOUT = 0
- WM_ERROR = 0
- WM_ERR_FLAG = 0
- ERRB = 1

WM is enabled with WM error (frozen frame)

- WM_DETOUT = 1
- WM_ERROR = 1
- WM_ERR_FLAG = 1
- ERRB = 0

Error condition occurs (INTR5 is read)

- WM_DETOUT = 1
- WM_ERROR = 1
- WM_ERR_FLAG = 0
- ERRB = 1

Note: When an error condition occurs and register INTR5 is read, WM_ERR_FLAG is cleared on read, but WM_ERROR remains high.

Error condition resolves (INTR5 is read)

- WM_DETOUT = 1
- WM_ERROR = 0
- WM_ERR_FLAG = 0
- ERRB = 1

Note: When an error condition goes away, WM_ERR_FLAG is cleared on read and WM_ERROR returns low.

Error condition resolves then returns

- WM_DETOUT = 1
- WM_ERROR = 1
- WM_ERR_FLAG = 1
- ERRB = 0

Note: When an error condition goes away and returns, WM_ERR_FLAG will be re-triggered high until it is read again.

Analog Confidential for NewVision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

17.3.3 Watermark Generation/Detection Test

The following procedure is recommended to test watermark operation.

- Check normal operation.
  - Configure the watermark generator and detectors as desired, and check the WM_DETOUT and WM_ERROR status registers, as well as checking that GPIO2 reflects the WM_DETOUT status as expected (if using WM_DETOUT output to GPIO2 mode)
- Check Error Handling
  - Configure the watermark generator as described in Watermark Generator (with Error Generation) to replicate a frozen frame condition by only generation two watermark and ensure the system error handling is correct. The WM_ERROR register should return a value of 1, and depending on configuration the ERRB pin, should become active and/or the video output should be blanked.
- Check Error Recovery
  - Configure the watermark generator back into normal.

17.3.4 Filter the Watermark Error Condition

The watermark detector can filter a watermark error condition. WMtimer[7:0] defines the time a watermark error condition must exist before an error is asserted, and is configured in 2ms steps that the frozen frame condition must exist before an error is triggered. A value of 0 (default) disables this filtering.

17.3.5 Watermark Operation Troubleshooting

If watermark generation/detection is not functioning properly:

- Verify that video data is being routed properly. Ensure that PCLKDET = 1 in pipe X of the serializer and VIDEO_LOCK = 1 in pipe X of the deserializer.
- Confirm that HS is available to embed watermark. If there is no horizontal sync (MIPI serializer), the Video Timing Generator (VTG) must be configured to generate a horizontal sync signal.
- Ensure that sync signal polarity is correct. Refer to the section below for how to determine proper sync polarity.

ADI Confidential for review under NDA
Analog Devices Confidential Distribution only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

17.3.6 Sync Polarity in Watermarking Blocks

The watermark block uses the vertical and horizontal sync signals to determine row and frame boundaries. The watermark block requires that both the HS and VS polarities be positive to operate correctly. A polarity control can be used to invert the sync signal internal to the watermark generator and detector while preserving the original sync polarity outside the watermark blocks.

1. VsyncPol:
    a. 0 (default) – retains sync signal polarity (use for positive sync polarity: 1'b1 during vertical blanking interval).
    b. 1 – inverts sync signal polarity (use for negative sync polarity: 0 during vertical blanking interval).
2. HsyncPol:
    a. 0 (default) – retains sync signal polarity (use for positive sync polarity: 1'b1 during vertical blanking interval).
    b. 1 – inverts sync signal polarity (use for negative sync polarity: 0 during vertical blanking interval).

Note: Sync signal polarities are application-specific (e.g., if the VESA guidelines are followed or a custom scheme is used). Some customers define the video timing and sync polarity throughout the telematic system. The ANSI/CTA-861-F standard is a suggested reference resource.

17.3.6.1 Searching for the Correct Watermark Polarity

If the correct sync polarities are unknown, the watermark generator/detector can be used to determine the correct sync signal polarities. Configure the generator to output an error condition and the detector to blank on watermark error, then cycle through the 16 possible polarity configurations (in generator and detector). When the screen blanks, note the active configuration as this is the correct polarity settings.

Generator Settings

Unlock registers:	WM_WREN_L = 0xBA
	WM_WREN_H = 0xDC
Enable the generator:	WM_EN = 1'b1
Enable ERR mode:	WM_MODE[2:0] = 3'b001
Lock registers:	WM_WREN_L = 0x00
	WM_WREN_H = 0x00

Detector Settings

Unlock registers:	WM_WREN_L = 0xBA
	WM_WREN_H = 0xDC
Configure as Detector:	WM_DET[1:0] = 2'b01
Enable the Detector:	WM_EN = 1'b1
Set the threshold correctly:	WM_TH[6:0] = 0x0E
Enable Video blanking:	WM_MASKMODE[1:0] = 2'b01
Lock registers:	WM_WREN_L = 0x00
	WM_WREN_H = 0x00

Loop through sync signal polarity combinations. A blank screen indicates that the sync signal polarity combination has been achieved.

API Confidential for review under NDA
Maxim Integrated Confidential, Provided to GMSL 2 Customers Under NDA

17.4 Feature Availability by Device Family

Watermarking is available to all GMSL2 devices (except Advanced CSI-2 Serializers). See Table 45 below for feature availability and the watermarking pipe.

Table 45. Watermarking Support by Device Family

Device Family	WM Generator/Detector	WM PIPE*
HDMI Serializers	X	Pipe X
Advanced HDMI Serializers	X	Pipe X
DSC Serializers	X	Pipe X
Advanced DSI Serializers	X	Pipe X
CSI-2 Serializers	X	Pipe X
Advanced CSI-2 Serializers		
CSI-2 Camera Deserializers	X	Pipe X
CSI-2 Display Deserializers	X	Pipe X
OLDI Deserializers	X	Pipe X
eDP/DP Deserializers	X	Pipe X
CSI-2 Quad Deserializers	X (2)	Pipe 0 and Pipe 4

Note: See the *Video Pipes* section for additional details.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Limited
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

18 Video Timing Generator (VTG) and Video Pattern Generator (VPG)

18.1 Overview

The Video Timing Generator (VTG) feature allows users to manipulate video timing parameters. The VTG generates or adjusts video sync signals to modify the timing details of incoming video streams or create video timing to be used with the Video Pattern Generator (VPG) function to generate test images. All GMSL2 serializers and the quad deserializers are equipped with these two features (refer to Feature Availability by Device Family section). The VTG and VPG are implemented within the video pipes on both the serializers and deserializers.

The VTG provides user-configurable options for the video sync signals: Vertical Sync (VS), Horizontal Sync (HS), and Data Enable (DE). When enabled, the VTG regenerates the video sync signals in accordance with user-defined timing parameters. These parameters offer flexibility to customize the sync polarity, pulse width, and timing of the regenerated video sync signals.

The VPG can generate user-defined RGB888 color patterns using the incoming PCLK and timing parameters sourced from either the input video or the VTG. These generated patterns are can be used to test the GMSL link and downstream devices (e.g., displays) without a video source connected to the serializer. GMSL2 serializers output these patterns to deserializers as standard video over the GMSL link (Figure 42); the GMSL2 deserializers that support VPG can output test patterns in MIPI CSI-2 format directly. Two types of patterns (i.e., color gradient and checkerboard) with configurable color parameters are available.

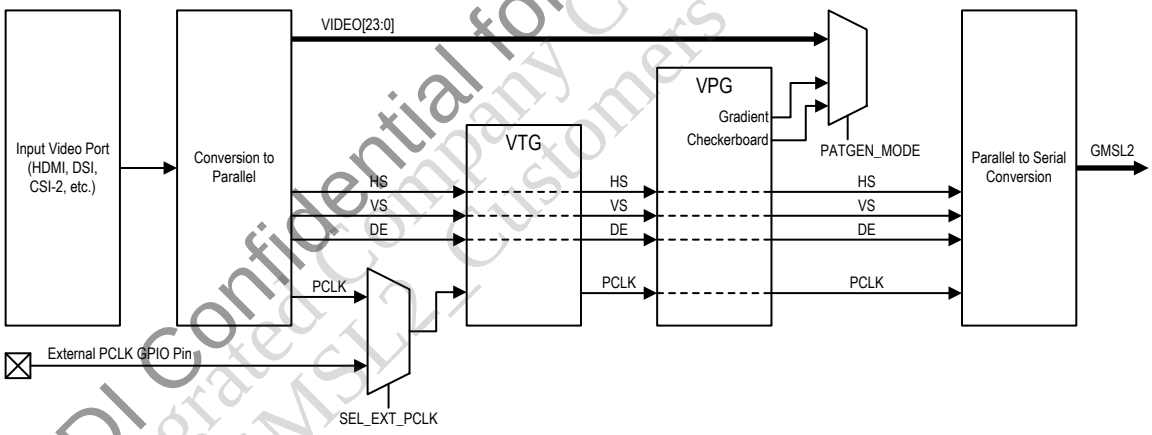


Figure 42. Video Path Through the VTG and VPG in GMSL2 Serializers

ADI Confidential for Review under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

18.2 Video Timing Generator (VTG)

18.2.1 VTG Operation

The core function of the VTG block is to generate VS, HS, and DE signals based off a trigger. The VTG can be configured to generate these signals internally or use the VS/HS/DE signals at the input of the VTG to drive the output signals (when disabled). This selection is made individually for each sync signal via the GEN_VS, GEN_HS, and GEN_DE register bits.

The VTG can be triggered by either a VS input transition (i.e., the external trigger) or an internally generated VS trigger (i.e., the tracking VS signal). In the case of the external trigger, the VS transition trigger is selected to be either the rising edge or the falling edge of the input VS signal with the VS_TRIG bit. Note that the selected edge for the input transition is referred to as the active edge.

The VTG consists of three pulse generators to generate the VS, HS, and DE signals. The polarity, start timing (i.e., delay from the trigger), periodicity, duty-cycle, and the number of HS and DE pulses per frame are all programmable.

Note: After the VTG is enabled, the first and/or second frame sync output pulses (VS/HS/DE) may be invalid.

18.2.2 VTG Configuration

Configuring the VTG consists of two steps: selecting the VTG operation mode (Table 46) and configuring the timing parameters for the VS, HS, and DE generation (Table 49).

Table 46. VTG Operation Mode Configuration Registers

Parameter	Register	Bitfield	Decode and Description
VS Generation Enable	VTX0	GEN_VS	0: Bypass VTG and use the VS signal from the video input 1: Generate VS signal from the VTG with specified timing
HS Generation Enable	VTX0	GEN_HS	0: Bypass VTG and use the HS signal from the video input 1: Generate HS signal from the VTG with specified timing
DE Generation Enable	VTX0	GEN_DE	0: Bypass VTG and use the DE signal from the video input 1: Generate DE signal from the VTG with specified timing
VS trigger mode	VTX0	VTG_MODE	See *VTG Trigger Modes* section below for information on following modes: 00: VS tracking mode 01: VS one-trigger mode 10: Auto-repeat mode 11: Free-running mode (default)
Select PCLK source	VTX1	SEL_EXT_PCLK	Selects the PCLK source for the VTG block. Reference *PCLK Selection* section below for details. 0: Use the PCLK received from the video interface input 1: Use an external PCLK applied to a GPIO Note: If using external PCLK mode with CSI-2 or DSI serializers, please refer to the *PCLK Selection* section below for additional configuration information.
VS trigger polarity	VTX1	VS_TRIG	0: Falling edge 1: Rising edge (default)

ADI Confidential for New vision under NDA
Maxim Integrated Confidential, Distribution Only Under NDA
Provided to GMSL2_ Customers Under NDA

Sync signal polarity	VTX0	VS_INV	The VTG timing configuration instructions assume positive sync polarity (sync pulse active high), so if negative VS polarity is desired, use this bit to invert. 0: Do not invert VS signal 1: Invert VS signal Note: This bit is active even when GEN_VS=0 and can be used to invert the VS polarity without configuring the VTG timing parameters.
Sync signal polarity	VTX0	HS_INV	The VTG timing configuration instructions assume positive sync polarity (sync pulse active high), so if negative HS polarity is desired, use this bit to invert. 0: Do not invert HS signal 1: Invert HS signal Note: This bit is active even when GEN_HS=0 and can be used to invert the HS polarity without configuring the VTG timing parameters.
Sync signal polarity	VTX0	DE_INV	The VTG timing configuration instructions assume positive sync polarity (sync pulse active high), so if negative DE polarity is desired, use this bit to invert. 0: Do not invert DE signal 1: Invert DE signal Note: This bit is active even when GEN_DE=0 and can be used to invert the DE polarity without configuring the VTG timing parameters.

18.2.2.1 VTG Trigger Modes

There are four different VTG trigger modes:

- **VS tracking mode** – This mode is used to reduce the glitches and jitters of the input VS signal. In this mode, the input VS period (VS_HIGH + VS_LOW) is tracked. After the VS tracking has locked, any VS input edge not in the expected PCLK cycle (e.g., glitch) is ignored. VS tracking is locked upon three consecutive periodic matches; VS tracking is unlocked following three consecutive periodic mismatches. At power-up or if VS tracking is unlocked, the next VS input edge is assumed to be the correct VS edge.
- **VS one-trigger mode** – In this mode, only one frame of VS, HS, and DE output signals is generated per VS input trigger. The polarity, timing (delay from the VS input trigger), and period/duty-cycle of the generated VS, HS, and DE signals are in accordance with the user-programmed parameters.
- **Auto-repeat mode** – This mode uses the VS input trigger to generate VS, HS, and DE signals as with VS one-trigger mode. However, instead of one frame per VS input trigger, auto-repeat mode generates continuous frames of VS, HS, and DE output signals following a VS input trigger. If the next VS input edge occurs earlier or later than expected by the VS period (VS_HIGH + VS_LOW), the newly generated frame will be considered correct. The previous VS/HS/DE signals will be cut or extended at the time point of the rising edge of the newly generated VS, HS, and DE signals.
- **Free-running mode (default)** – This mode is based on auto-repeat mode. In this mode, the VS input signal is not needed to generate continuous frames of VS, HS, and DE output signals. The VTG automatically starts generating a continuous stream of frames, consisting of VS, HS, and DE signals in accordance with the user-programmed parameters.

Maxim Integrated Confidential, Provided to GMSL Customers Under NDA
API Confidential for New Version under NDA

18.2.2.2 PCLK Selection

There are three options to provide a PCLK to the VTG block: using the clock from the video input, supplying an external PCLK via a GPIO, or using an internally generated PCLK (GMSL2 quad deserializers only).

The correct PCLK value is important for achieving the desired pattern resolutions and frame rates. Reference the Video Basics section for more information pertaining to calculating the PCLK value. Note that supplying an external PCLK via GPIO offers the most precise control of the PCLK value.

18.2.2.2.1 Clock from the Input Video Source

The PCLK is calculated differently depending on the video interface:

- **HDMI serializers in HDMI 1.4 mode**

$$PCLK = HDMI\ Clock$$

- **HDMI serializers in HDMI 2.0 mode**

$$PCLK = \frac{HDMI\ Clock}{4}$$

- **DSI serializers**

$$PCLK = \frac{Data\ rate * Lanes}{bpp}$$

where data rate is double the MIPI clock rate, lanes is the number of active MIPI lanes, and bpp is bits per pixel (which is always equal to 24 for DSI applications).

- **CSI-2 Serializers**

$$PCLK = \frac{Data\ rate * Lanes}{bpp}$$

where data rate is double the MIPI clock rate, lanes is the number of active MIPI lanes, and bpp is bits per pixel (which is determined by the CSI-2 datatype).

18.2.2.2.2 External Clock via GPIO

HDMI serializers, advanced HDMI serializers, and CSI-2 serializers have a GPIO pin that can be used as a PCLK input pin. An external clock can be injected directly on this GPIO to trigger the VTG. This GPIO varies by device family as given in *Feature Availability by Device Family* section.

To use the GPIO input with a CSI-2 serializer, the device must be programmed into parallel mode so that the GPIO 0 pin becomes the PCLK input pin. With this configuration, only the pattern generator in video pipe X can be used. Enable parallel mode by setting bit 0 of register PAR_VID_EN in REG7.

In most cases, a camera serializer can output a reference clock from another GPIO to eliminate the need of an external clock source. The relevant registers for configuration are presented in *Table 47*.

Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA
Analog Devices Confidential Distribute only under NDA
ADI Confidential for Newvision under NDA

Table 47. Additional CSI-2 Serializer PCLK Register Settings

Register	Bits	Default Value	Description
REG7 (0x0007)	7:0	0xF6	**Bit [7:4]: Parallel Data Input Pins:** Parts dependent; Refer to DS/Reg map for details. **Bit 2: Parallel VS Input Pin:** 0: Parallel VS Input disabled 1: Parallel VS Input enabled **Bit 1: Parallel HS Input Pin:** 0: Parallel HS Input disabled 1: Parallel HS Input enabled **Bit 0: Parallel Video Input:** 0: Parallel video disabled 1: Parallel video enabled
REF_VTG0 (0x03F0)	1	0	**Bit 0: Reference PLL enable:** 0: Disable reference generation PLL 1: Enable reference generation PLL
REF_VTG0 (0x03F0)	5:4	01	**Bit [5:4]: Reference PLL frequency setting:** 00: 19.2MHz 01: 27MHz 10: 37.125MHz 11: 74.25MHz
REF_VTG1 (0x03F1)	0	0	**Bit 0: PCLKEN:** 0: PCLK output disabled 1: PCLK output enabled on selected PCLK_GPIO
REF_VTG1 (0x03F1)	5:1	00000	**Bit [5:1]: PCLK_GPIO:** Select the reference clock output GPIO

18.2.2.2.3 Using an Internally Generated PCLK

GMSL2 quad deserializers have two clock configuration registers for setting the PCLK value for the video pipes. The video pattern PCLK frequency can optionally be set to 25 or 75MHz for all eight pipes or 150 or 375MHz on a per pipe basis. See Video Pattern PLCK Settings Registers (*Table 48*) for configuration details.

Table 48. Video Pattern PCLK Selection

Register Name (Reg #)		PCLK Frequency
DEBUG_EXTRA (0x09 [1:0])	PATGEN_CLK_SRC (bit 7)	
00	X	25MHz
01	X	75MHz
1x	0	150MHz
1x	1	375MHz

ADI Confidential for NEW vision under NDA
Analog Devices Confidential Only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

18.2.3 VTG Timing Parameters

The sync pulse timing parameters for the VTG are shown in *Figure 43* and described in *Table 49*.

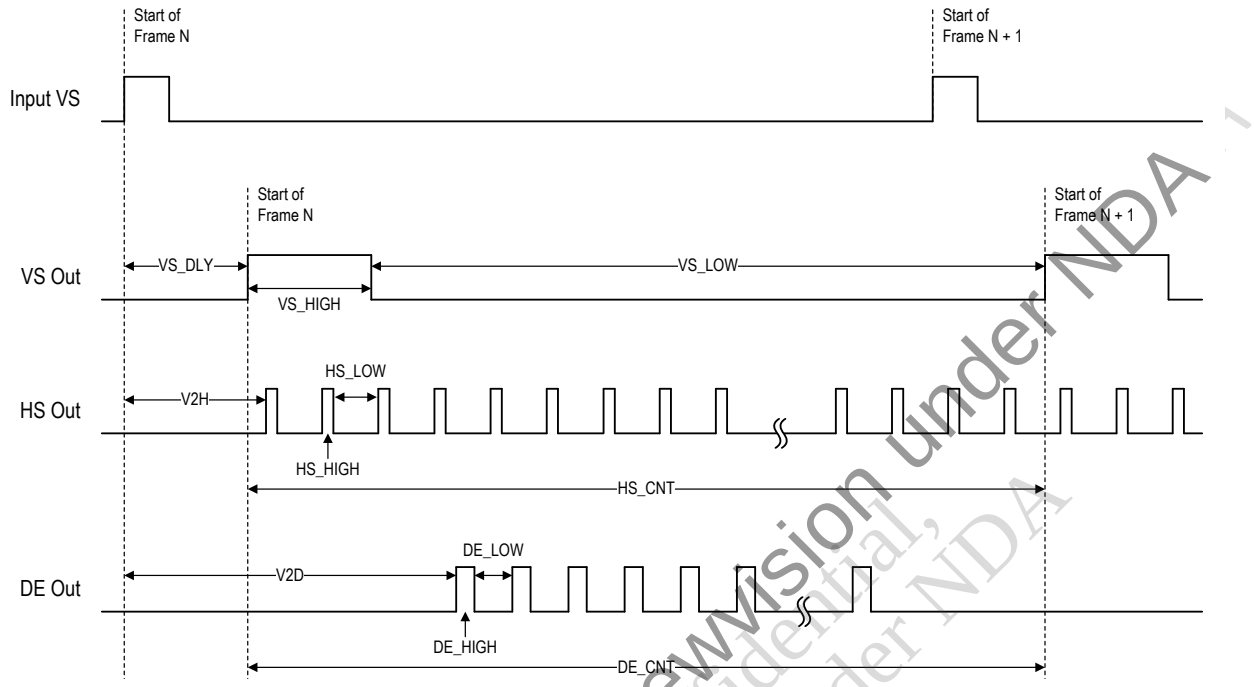


Figure 43. VTG Timing parameters

All parameters are defined in terms of PCLKs. Vertical timing parameters must be converted from lines to pixel clocks by multiplying by H_{total}.

Only the parameters associated with the sync pulses that are enabled must be configured. For example, if only VS is being generated by the VTG (GEN_VS=1, GEN_HS=0, GEN_DE=0), only VS_DLY, VS_HIGH, and VS_LOW must be configured.

Note: In cases where the VTG is used to regenerate DE in combination with incoming video data, the incoming data is not retimed to DE if it is adjusted from the original timing. This will result in lost video data. E.g., If DE is delayed four PCLKs, the first four pixels from the incoming data will be lost and the last four will converted to zeros (i.e., padded).

ADI Confidential for Newvision under NDA
Analog Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

Table 49. Timing Parameter Configuration Registers

Parameter	Register	Bitfield	Description
VS_DLY	VTX2 VTX3 VTX4	VS_DLY_2[7:0] VS_DLY_1[7:0] VS_DLY_0[7:0]	The delay from the VS trigger to the generated VS signal in terms of PCLKs. Note, if using the input video stream, this will delay the output sync signals relative to the video data.
VS_HIGH	VTX5 VTX6 VTX7	VS_HIGH_2[7:0] VS_HIGH_1[7:0] VS_HIGH_0[7:0]	The high duration of the generated VS output signal in terms of PCLKs.
VS_LOW	VTX8 VTX9 VTX10	VS_LOW_2[7:0] VS_LOW_1[7:0] VS_LOW_0[7:0]	The low duration of the generated VS output signal in terms of PCLKs.
V2H	VTX11 VTX12 VTX13	V2H_2[7:0] V2H_1[7:0] V2H_0[7:0]	The delay from the VS trigger to the rising edge of the generated HS signal.
HS_HIGH	VTX14 VTX15	HS_HIGH_1[7:0] HS_HIGH_0[7:0]	The high duration of the generated HS output signal in terms of PCLKs.
HS_LOW	VTX16 VTX17	HS_LOW_1[7:0] HS_LOW_0[7:0]	The low duration of the generated HS output signal in terms of PCLKs.
HS_CNT	VTX18 VTX19	HS_CNT_1[7:0] HS_CNT_0[7:0]	The number of HS output pulses generated per video frame.
V2D	VTX20 VTX21 VTX22	V2D_2[7:0] V2D_1[7:0] V2D_0[7:0]	The delay from the VS trigger to the rising edge of the generated DE signal.
DE_HIGH	VTX23 VTX24	DE_HIGH_1[7:0] DE_HIGH_0[7:0]	The high duration of the generated DE output signal in terms of PCLKs.
DE_LOW	VTX25 VTX26	DE_LOW_1[7:0] DE_LOW_0[7:0]	The low duration of the generated DE output signal in terms of PCLKs.
DE_CNT	VTX27 VTX28	DE_CNT_1[7:0] DE_CNT_0[7:0]	The number of DE pulses generated per video frame.

Note: For all VTG parameters that span multiple registers, the MSBs are programmed into the first register.

18.2.3.1 Programming for Standard Video Timings

The VTG should be configured to match the timing specification of the target application. Though, in many cases, it is preferred to use standard video timings. The video timings for standard resolutions are defined by the CTA-861 specification. The specification also defines the relationship of the sync signals and stipulates, in particular, that the active edge of VS and HS be synchronous. This definition is what will be used in the remainder of this section; however, it is important to note that some displays and applications may define these timings differently.

ADI Confidential for New Version under NDA
Analog Devices Confidential, Under NDA
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2 Customers

For a standard video with timings as defined in *Figure 44*, the recommended values to program into the VTG are listed in *Table 50*.

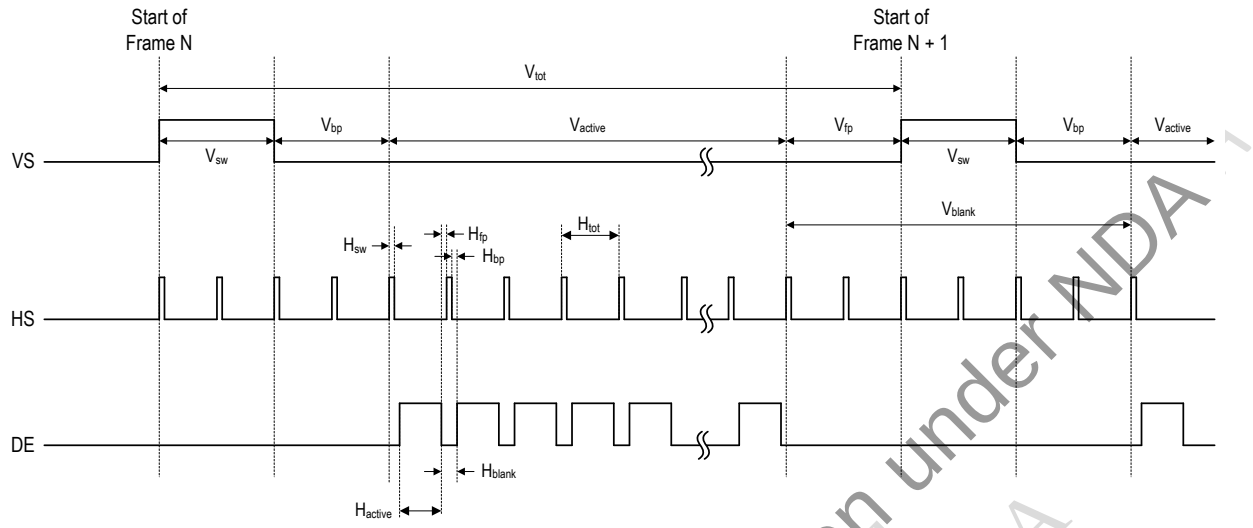


Figure 44. Video Timing Definition

The diagram in *Figure 44* shows a vertical front porch, sync width, and back porch all equal to two lines. Note that the HS and VS rising edges are synchronous. *Figure 45* below illustrates the relationship between the different components of the HS and DE signals.

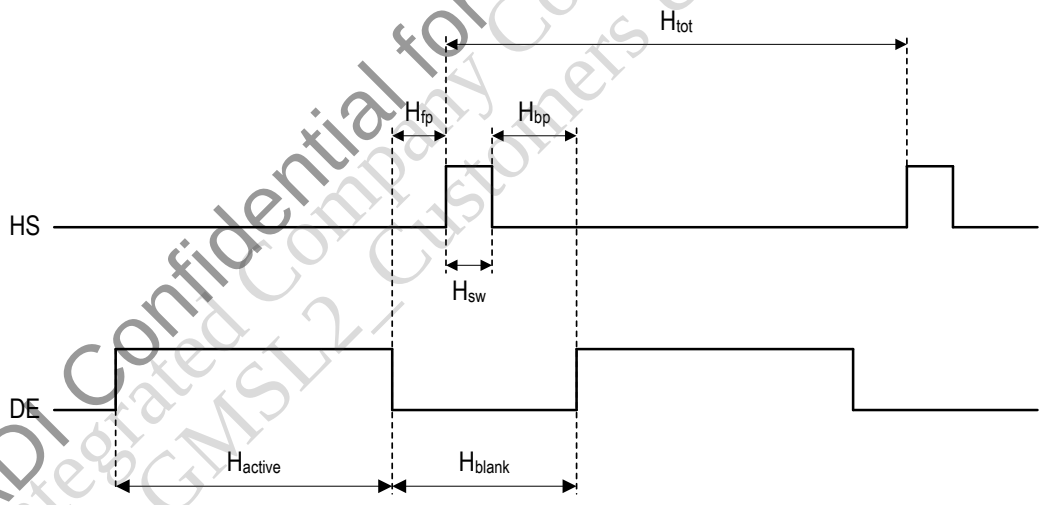


Figure 45. HS and DE Signal Components

ADI Confidential for New Vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

Table 50. Recommended Register Settings for VTG Parameters (Standard Video Timings)

Parameter	Recommended Value	Recommended Value Description
VS_DLY	0	Setting VS delay will cause the output frame to not be delayed with respect to the input trigger. If a delay is desired, program this register with the delay time terms of PCLK periods. A delay will offset the output sync pulse frame with respect to the input frame.
VS_HIGH	V_{sw} * H_{tot}	Program to V_{sw} (multiplied by H_{tot} to convert from lines to PCLK cycles).
VS_LOW	(V_{active} + V_{fp} + V_{bp}) * H_{tot}	Program to the remainder of the vertical period: V_{tot} - V_{sw}. This is equivalent to V_{active} + V_{fp} + V_{bp} (multiplied by H_{tot} to convert from lines to PCLK cycles).
V2H	VS_DLY	This should be set equal to VS_DLY to align the rising edges of the generated VS and HS pulses (as specified in the CTA-861 specification). If VS_DLY is set to 0, V2H should also be set to 0.
HS_HIGH	H_{sw}	Program to H_{sw}
HS_LOW	H_{active} + H_{fp} + H_{bp}	Program to the remainder of the horizontal period: H_{tot} - H_{sw}. This is equivalent to H_{active} + H_{fp} + H_{bp}
HS_CNT	V_{tot}	The number of HS pulses needed is equal to the number of total lines (active plus blanking since HS are present through the vertical blanking period)
V2D	VS_DLY + H_{tot} * (V_{sw} + V_{bp}) + (H_{sw} + H_{bp})	V2D defines the delay from the active edge of the VS trigger to the rising edge of the first DE signal in a frame. To calculate, begin by aligning with the start of the generated frame by adding the VS_DLY value. Then, add the vertical sync width and the vertical back porch time in terms of PCLK cycles (V_{sw} + V_{bp} multiplied by H_{tot}). Since the first active line begins with a HS signal, add the horizontal sync width and the horizontal back porch. This sum is the delay to the first DE rising edge.
DE_HIGH	H_{active}	Program with H_{active} so that DE is the active line length.
DE_LOW	H_{fp} + H_{sw} + H_{bp}	Program with the horizontal blanking period (H_{blank}), which is equal to H_{fp} + H_{sw} + H_{bp}
DE_CNT	V_{active}	The number of DE pulses needed in a frame is equal to the number of active lines (there are no DE pulses in the vertical blanking period).

ADI Confidential for New Version under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA

18.2.3.2 VTG Programming Example

The following example script is for HDMI serializers and video with VESA-standard 1080p resolution with timing:

PCLK = 148.5MHz
$[H_{tot}, H_{active}, H_{fp}, H_{sw}, H_{bp}] = [2200, 1920, 88, 44, 148]$
$[V_{tot}, V_{active}, V_{fp}, V_{sw}, V_{bp}] = [1125, 1080, 4, 5, 36]$

```
#Enable the VTG for 1920x1080p video and do not modify the timing
0x80,0x01C8,0xE1
0x80,0x01C9,0xE1
0x80,0x01CA,0x00
0x80,0x01CB,0x00
0x80,0x01CC,0x00
0x80,0x01CD,0x00
0x80,0x01CE,0x2A
0x80,0x01CF,0xF8
0x80,0x01D0,0x25
0x80,0x01D1,0x99
0x80,0x01D2,0x00
0x80,0x01D3,0x00
0x80,0x01D4,0x00
0x80,0x01D5,0x00
0x80,0x01D6,0x00
0x80,0x01D7,0x2C
0x80,0x01D8,0x08
0x80,0x01D9,0x6C
0x80,0x01DA,0x08
0x80,0x01DB,0x98
0x80,0x01DC,0x01
0x80,0x01DD,0x61
0x80,0x01DE,0x18
0x80,0x01DF,0x07
0x80,0x01E0,0x80
0x80,0x01E1,0x01
0x80,0x01E2,0x18
0x80,0x01E3,0x04
0x80,0x01E4,0x38
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

18.3 Video Pattern Generator (VPG)

18.3.1 VPG Operation

The VPG block generates and outputs video data based on specified timing parameters. The video timing is sourced from either the input video stream or user-configured timing values in the associated VTG block. The VPG outputs either a color gradient pattern or a checkerboard pattern with user-definable color parameters and pattern details.

The VPG allows users to perform video tests without a video source. Here, patterns generated by the serializer are sent through the serial link, received by the deserializer, and output to test the serial link and downstream devices such as displays. Patterns generated by the quad CSI-2 deserializer can output MIPI CSI-2 data directly without any connection to the serial link.

Note: There is a GUI tool that can be used to configure the VTG and VPG.

18.3.2 VPG Modes of Operation

The VPG has two available patterns with configurable options: color gradient and checkerboard. The color gradient pattern ramps across the full 8-bit range (0x00–0xFF) for blue, green, red, and white (all colors simultaneously) as depicted in *Figure 46*. The checkerboard pattern consists of two alternating colors (*Figure 47* shows blue and red); the colors and size of the pattern are configurable. Note that, since there is no frame memory, other patterns cannot be loaded into the GMSL devices.

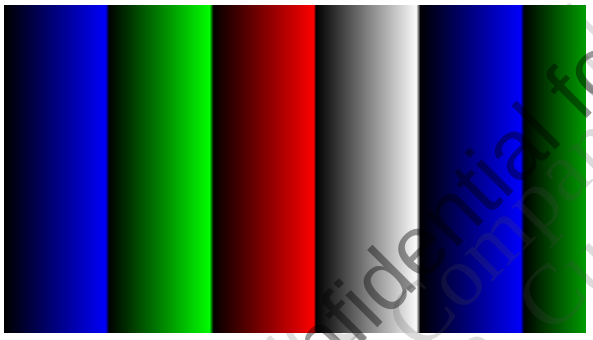


Figure 46. VPG – Gradient Pattern

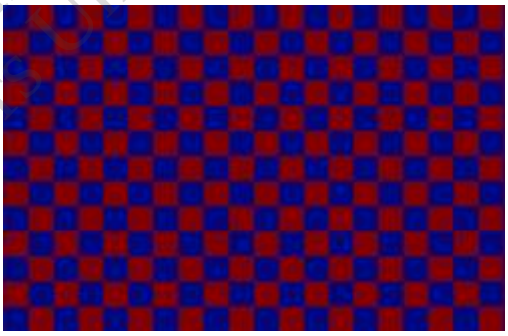


Figure 47. VPG – Checkerboard Pattern

ADI Confidential for New Version under NDA
Analog Devices Inc. Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_ Customer Under NDA

18.3.3 VPG Configuration

In gradient mode, the length of the gradient pattern and gradient direction are configurable. In checkerboard mode, two colors (i.e., color A and color B) are selected and the size of the pattern is user-defined. The first video line starts with color A for the programmed number of pixels, then switches to color B for its programmed length. After color B, the pattern repeats. After a set number of lines, the pattern swaps and outputs color B before color A. This process repeats for the height of the frame. The configuration registers for serializers are listed in *Table 51* and for the quad deserializers in *Table 52*. Operation is the same between devices; however, there are small differences in the bitfield names.

Table 51. VPG Configuration Registers for Serializers

Parameter	Register	Bitfield	Decode and Description
Select the pattern type	VTX29	PATGEN_MODE[1:0]	Select the VPG pattern. 00: Pattern generator disabled, use video from the serializer input (default) 01: Generate checkerboard pattern 10: Generate gradient pattern 11: Reserved
In gradient mode: Select the gradient direction	VTX29	GRAD_MODE[0]	0: Gradient mode increasing. Each gradient color starts from a value of 0x00 and increases to 0xFF 1: Gradient mode decreasing. Each gradient color starts from a value of 0xFF and decreases to 0x00
In gradient mode: Select the gradient pattern length	VTX30	GRAD_INC[7:0]	Selects the value each pixel increments, program to the desired increment amount multiplied by 4. The default value of 4 results in each pixel incrementing by 1, resulting in a pattern length of 256 pixels per color.
In checkerboard mode: Set the value of color A	VTX31 VTX32 VTX33	CHKR_A_L[7:0] CHKR_A_M[7:0] CHKR_A_H[7:0]	Sets the red component of color A Sets the green component of color A Sets the blue component of color A
In checkerboard mode: Set the value of color B	VTX34 VTX35 VTX36	CHKR_B_L[7:0] CHKR_B_M[7:0] CHKR_B_H[7:0]	Sets the red component of color B Sets the green component of color B Sets the blue component of color B
In checkerboard mode: Set the length of color A	VTX37	CHKR_RPT_A[7:0]	Sets the number of pixels of color A. The first line outputs color A first.
In checkerboard mode: Set the length of color B	VTX38	CHKR_RPT_B[7:0]	Sets the number of pixels of color B. The first line outputs color B after CHKR_RPT_A pixels. Set equal to CHKR_RPT_A for a square checkerboard pattern.
In checkerboard mode: Set the height of the checkerboard	VTX39	CHKR_ALT[7:0]	After CHKR_ALT lines, the pattern switches to output color B before color A. Set equal to CHKR_RPT_A and CHKR_RPT_B for a square checkerboard pattern.

ADI Confidential for New Version under NDA
Analog Devices Confidential
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

Table 52. VPG Configuration Registers for Quad Deserializers

Parameter	Register	Bitfield	Decode and Description
Select the pattern type	VTX29	PATGEN_MODE[1:0]	Select the VPG pattern. 00: Pattern generator disabled; use video from the serializer input (default) 01: Generate checkerboard pattern 10: Generate gradient pattern 11: Reserved
In gradient mode: Select the gradient direction	VTX29	GRAD_MODE[0]	0: Gradient mode increasing. Each gradient color starts from a value of 0x00 and increases to 0xFF 1: Gradient mode decreasing. Each gradient color starts from a value of 0xFF and decreases to 0x00
In gradient mode: Select the gradient pattern length	VTX30	GRAD_INCR[7:0]	Selects the value each pixel increments, program to the desired increment amount multiplied by 4. The default value of 4 results in each pixel incrementing by 1, resulting in a pattern length of 256 pixels per color.
In checkerboard mode: Set the value of color A	VTX31 VTX32 VTX33	CHKR_COLOR_A_L[7:0] CHKR_ COLOR_A_M[7:0] CHKR_ COLOR_A_H[7:0]	Sets the red component of color A Sets the green component of color A Sets the blue component of color A
In checkerboard mode: Set the value of color B	VTX34 VTX35 VTX36	CHKR_ COLOR_B_L[7:0] CHKR_ COLOR_B_M[7:0] CHKR_ COLOR_B_H[7:0]	Sets the red component of color B Sets the green component of color B Sets the blue component of color B
In checkerboard mode: Set the length of color A	VTX37	CHKR_RPT_A[7:0]	Sets the number of pixels of color A. The first line outputs color A first.
In checkerboard mode: Set the length of color B	VTX38	CHKR_RPT_B[7:0]	Sets the number of pixels of color B. The first line outputs color B after CHKR_RPT_A pixels. Set equal to CHKR_RPT_A for a square checkerboard pattern.
In checkerboard mode: Set the height of the checkerboard	VTX39	CHKR_ALT[7:0]	After CHKR_ALT lines, the pattern switches to output color B before color A. Set equal to CHKR_RPT_A and CHKR_RPT_B for a square checkerboard pattern.

18.3.3.1 Application Setup of VPG

The following is a typical procedure to enable the VPG for serializers:

- Connect the deserializer and program it for the desired output format(s).
- Program the VTG timing details (i.e., resolution, blanking, and frame rate) and calculate required PCLK frequency.
- Program the VPG pattern details (i.e., pattern type and colors).
- Feed the PCLK and enable VTG/VPG via registers.

For additional information on configuring VPG in the quad deserializers, refer to the Video Pattern Generator (VPG) section in the quad deserializer section.

Analog Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

18.4 Feature Availability by Device Family

VTP and VPG support by device family is presented in Table 53.

Table 53. VTG/VPG Support by Device Family

Device Family	VTG Available	External PCLK GPIO Pin	VPG Available
HDMI Serializers	Pipe X	GPIO 4	Pipe X
Advanced HDMI Serializers	Pipe X	GPIO 4	Pipe X
DSI Serializers	All four video pipes	No	All four video pipes
Advanced DSI Serializers	All four video pipes	No	All four video pipes
CSI-2 Serializers	All four video pipes	GPIO 0	All four video pipes
Advanced CSI-2 Serializers	All four video pipes	No	All four video pipes
CSI-2 Camera Deserializers	No	No	No
CSI-2 Display Deserializers	No	No	No
OLDI Deserializers	No	No	No
eDP/DP Deserializers	No	No	No
CSI-2 Quad Deserializers	All eight video pipes	No	Two available.

Note: Reference the CSI-2 Quad Deserializer section for information regarding VTG and VPG implementation.

ADI Confidential for New vision under NDA
Analog Devices Confidential Distributed by Maxim Integrated Company Confidential Provided to GMSL2_Customers Under NDA

19 Video Crossbar

19.1 Overview

GMSL2 devices integrate video crossbar switches that allows the reordering color and sync signals. This enables the serial link to be configured as a bridge between different input and output video interfaces and ensures wide format compatibility. Video bits can be selected to be inverted or forced to either 1 or 0 states. Video crossbar is used between standardized input and output video interfaces (e.g., CSI-2, DSI, and HDMI) that accept only one possible bitmapping format. It is also used for interfaces with nonstandard bitmapping or interfaces with multiple bitmaps (e.g., Parallel or LVDS). All GMSL2 devices (except eDP/DP deserializers) contain video crossbar switches to ensure compatibility even if the companion device does not (e.g., a GMSL1 device).

19.2 Operation

The GMSL2 video crossbar switches allow any input video bit to be mapped to any output video bit via register settings. It is also possible to configure one video crossbar input to multiple outputs. Each video crossbar output has dedicated bitmap configuration registers (Table 54).

Table 54. Video Crossbar Output Bitmapping Registers

Bitfield	Bit Position	Default Value	Description
CROSSx__I	6	0	Enable Crossbar Bit Inversion: Inverts this video bit when set to 1. If this bit and the force bit is enabled at the same time, the video bit will be forced high
CROSSx__F	5	0	Enable Crossbar Bit Forcing: Forces this video bit low when set to 1. If this bit and the invert bit is enabled at the same time, the video bit will be forced high
CROSSx_	[4:0]	See description	Crossbar Input Selection: Select the input bit to map to the output bit. By default, CROSS0 is mapped from input bit 0, CROSS1 is mapped from input bit 1 etc.

ADI Confidential for review under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers under NDA

Figure 48 below illustrates the video crossbar block diagram for a single output bit.

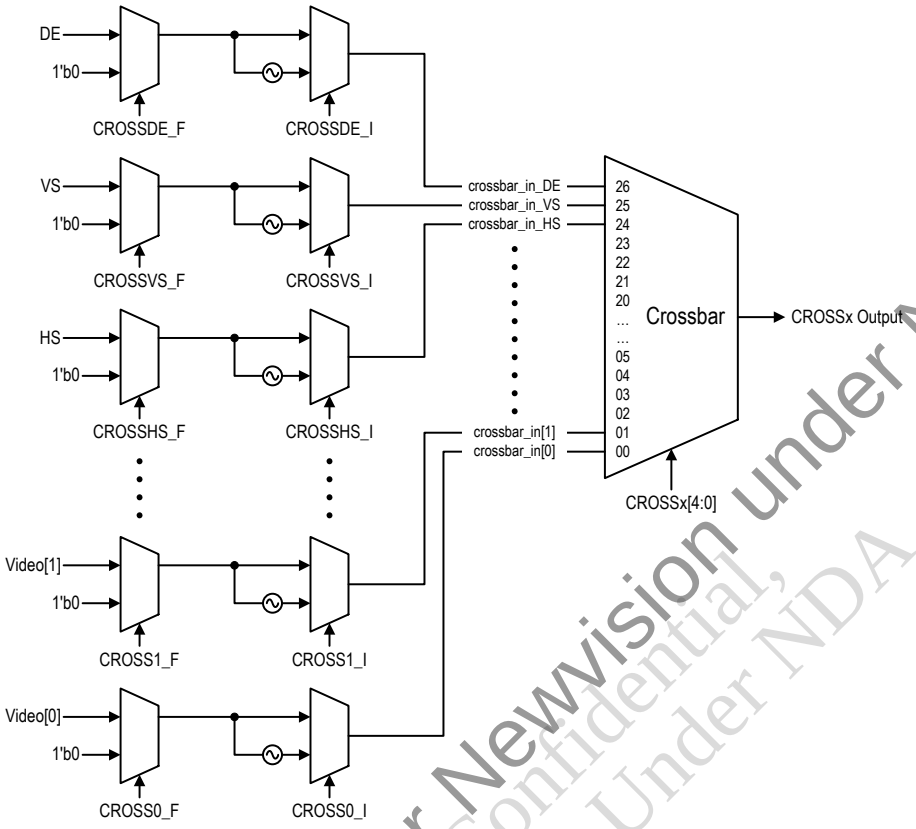


Figure 48. Video Crossbar Block Diagram (Single Output Bit)

19.3 Configuration

The video crossbar switches should be configured prior to initializing video on the serial link system. Ensure that the video source hardware, serial link, and display hardware are properly connected. Then, configure the video crossbar registers according to the colormaps of the input and output video formats of the application. In the video crossbar registers, confirm that the color bits are reordered and/or rerouted as needed and adhere to the relevant colormap tables. Start the video transmission, wait for video lock, and verify the functionality of the video crossbar. Reference the Status and Debug Registers section if undesired behavior is exhibited.

19.3.1 Video Crossbar Registers for Different Interfaces

Each video pipe has dedicated video crossbar switches with independent configuration registers. In cases with multiple video pipes, the video crossbar register names are appended with extensions (i.e., _X, _Y, _Z, and _U) that correspond with the assigned video pipe.

The CROSS_[23:0] registers contain video data crossbar settings and are common to all video interfaces.

Interface-specific configuration information is provided below.

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Provided to CMSL's Customers Under NDA

19.3.1.1 HDMI Input

The VTX[42:40] registers configure to the DE, VS, and HS signals, respectively. GMSL2 uses an active-high VS; for active-low resolutions (e.g., STMPE resolutions), invert the VS signal in the video crossbar switch.

19.3.1.2 CSI/DSI and Parallel Input

The VTX[42:40] registers configure the DE, VS, and HS signals, respectively.

19.3.1.2.1 CSI/DSI Mode

The start of a long packet is mapped to a DE rising edge, and the end of a long packet is mapped to a DE falling edge. The frame start packet is mapped to a VS rising edge, and the frame end packet is mapped to a VS falling edge. CSI-2 ignores and does not map HS signals. In DSI mode, however, the line start packet is mapped to a HS rising edge, and the line end packet is mapped to a HS falling edge.

19.3.1.2.2 Parallel Mode

Video sync signals typically consist of HS/VS/DE; however, some systems only send HS/VS (the Data Enable signal is labelled as “HS” and the horizontal sync signal does not exist). In this case, set the DE crossbar switch to use the HS input. Configure the invert bit as needed to ensure that DE is fed an active-high signal. GMSL2 uses an active-high VS; if VS is active-low (e.g., STMPE resolutions), VS should be inverted in the crossbar switch.

19.3.1.3 OLDI Output

The CROSS_[27:24] registers configure the LVDS RES, DE, VS, and HS signals, respectively.

19.3.1.4 CSI-2 Output

The CROSS_HS, CROSS_VS, and CROSS_DE registers configure the respective video sync signals. The start of a long packet is mapped to a DE rising edge, and the end of a long packet is mapped to a DE falling edge. The frame start packet is mapped to a VS rising edge, and the frame end packet is mapped to a VS falling edge. CSI-2 ignores and does not map HS signals. The CROSS_[29:27] registers are only used in GMSL1 backward compatible mode.

19.3.1.5 eDP/DP Output

eDP/DP devices do not contain video crossbar switches.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

19.3.2 Colormaps for Video Formats

Table 55. RGB Color Map

Input Bits	RGB888	RGB666	RGB565
DIN[0]	R[0]	R[0]	R[0]
DIN[1]	R[1]	R[1]	R[1]
DIN[2]	R[2]	R[2]	R[2]
DIN[3]	R[3]	R[3]	R[3]
DIN[4]	R[4]	R[4]	R[4]
DIN[5]	R[5]	R[5]	G[0]
DIN[6]	R[6]	G[0]	G[1]
DIN[7]	R[7]	G[1]	G[2]
DIN[8]	G[0]	G[2]	G[3]
DIN[9]	G[1]	G[3]	G[4]
DIN[10]	G[2]	G[4]	G[5]
DIN[11]	G[3]	G[5]	B[0]
DIN[12]	G[4]	B[0]	B[1]
DIN[13]	G[5]	B[1]	B[2]
DIN[14]	G[6]	B[2]	B[3]
DIN[15]	G[7]	B[3]	B[4]
DIN[16]	B[0]	B[4]	—
DIN[17]	B[1]	B[5]	—
DIN[18]	B[2]	—	—
DIN[19]	B[3]	—	—
DIN[20]	B[4]	—	—
DIN[21]	B[5]	—	—
DIN[22]	B[6]	—	—
DIN[23]	B[7]	—	—

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Table 56. RAW Color Map

Input Bits	RAW (8bit)	RAW (8bit) DBL	RAW (10bit)	RAW (10bit) DBL	RAW (12bit)	RAW (12bit) DBL	RAW (14bit)	RAW (16bit)	RAW (20bit)
DIN[0]	P[0]	P[0]	P[0]	P[0]	P[0]	P[0]	P[0]	P[0]	P[0]
DIN[1]	P[1]	P[1]	P[1]	P[1]	P[1]	P[1]	P[1]	P[1]	P[1]
DIN[2]	P[2]	P[2]	P[2]	P[2]	P[2]	P[2]	P[2]	P[2]	P[2]
DIN[3]	P[3]	P[3]	P[3]	P[3]	P[3]	P[3]	P[3]	P[3]	P[3]
DIN[4]	P[4]	P[4]	P[4]	P[4]	P[4]	P[4]	P[4]	P[4]	P[4]
DIN[5]	P[5]	P[5]	P[5]	P[5]	P[5]	P[5]	P[5]	P[5]	P[5]
DIN[6]	P[6]	P[6]	P[6]	P[6]	P[6]	P[6]	P[6]	P[6]	P[6]
DIN[7]	P[7]	P[7]	P[7]	P[7]	P[7]	P[7]	P[7]	P[7]	P[7]
DIN[8]	—	P[0]	P[8]	P[8]	P[8]	P[8]	P[8]	P[8]	P[8]
DIN[9]	—	P[1]	P[9]	P[9]	P[9]	P[9]	P[9]	P[9]	P[9]
DIN[10]	—	P[2]	—	P[0]	P[10]	P[10]	P[10]	P[10]	P[10]
DIN[11]	—	P[3]	—	P[1]	P[11]	P[11]	P[11]	P[11]	P[11]
DIN[12]	—	P[4]	—	P[2]	—	P[0]	P[12]	P[12]	P[12]
DIN[13]	—	P[5]	—	P[3]	—	P[1]	P[13]	P[13]	P[13]
DIN[14]	—	P[6]	—	P[4]	—	P[2]	—	P[14]	P[14]
DIN[15]	—	P[7]	—	P[5]	—	P[3]	—	P[15]	P[15]
DIN[16]	—	—	—	P[6]	—	P[4]	—	—	P[16]
DIN[17]	—	—	—	P[7]	—	P[5]	—	—	P[17]
DIN[18]	—	—	—	P[8]	—	P[6]	—	—	P[18]
DIN[19]	—	—	—	P[9]	—	P[7]	—	—	P[19]
DIN[20]	—	—	—	—	—	P[8]	—	—	—
DIN[21]	—	—	—	—	—	P[9]	—	—	—
DIN[22]	—	—	—	—	—	P[10]	—	—	—
DIN[23]	—	—	—	—	—	P[11]	—	—	—

Time 0
Time 1

ADI Confidential For New Version under NDA!
Analog Devices Confidential Distribution Unlimited
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers under NDA!

Table 57. YUV/YCrCB Color Map

Input Bits	YUV422 (8bit)	YUV422 (8bit) Mux=1	YUV422 (10bit)	YUV422 (10bit) Mux=1	YUV444 (8bit)	YUV422 (12bit)
DIN[0]	Cb/Cr [0]	YCb/YCr [0]	Cb/Cr [0]	YCb/YCr [0]	Cr [0]	Cb/Cr [0]
DIN[1]	Cb/Cr [1]	YCb/YCr [1]	Cb/Cr [1]	YCb/YCr [1]	Cr [1]	Cb/Cr [1]
DIN[2]	Cb/Cr [2]	YCb/YCr [2]	Cb/Cr [2]	YCb/YCr [2]	Cr [2]	Cb/Cr [2]
DIN[3]	Cb/Cr [3]	YCb/YCr [3]	Cb/Cr [3]	YCb/YCr [3]	Cr [3]	Cb/Cr [3]
DIN[4]	Cb/Cr [4]	YCb/YCr [4]	Cb/Cr [4]	YCb/YCr [4]	Cr [4]	Cb/Cr [4]
DIN[5]	Cb/Cr [5]	YCb/YCr [5]	Cb/Cr [5]	YCb/YCr [5]	Cr [5]	Cb/Cr [5]
DIN[6]	Cb/Cr [6]	YCb/YCr [6]	Cb/Cr [6]	YCb/YCr [6]	Cr [6]	Cb/Cr [6]
DIN[7]	Cb/Cr [7]	YCb/YCr [7]	Cb/Cr [7]	YCb/YCr [7]	Cr [7]	Cb/Cr [7]
DIN[8]	Y[0]	—	Cb/Cr [8]	YCb/YCr [8]	Y[0]	Cb/Cr [8]
DIN[9]	Y[1]	—	Cb/Cr [9]	YCb/YCr [9]	Y[1]	Cb/Cr [9]
DIN[10]	Y[2]	—	Y[0]	—	Y[2]	Cb/Cr [10]
DIN[11]	Y[3]	—	Y[1]	—	Y[3]	Cb/Cr [11]
DIN[12]	Y[4]	—	Y[2]	—	Y[4]	Y[0]
DIN[13]	Y[5]	—	Y[3]	—	Y[5]	Y[1]
DIN[14]	Y[6]	—	Y[4]	—	Y[6]	Y[2]
DIN[15]	Y[7]	—	Y[5]	—	Y[7]	Y[3]
DIN[16]	—	—	Y[6]	—	Cb [0]	Y[4]
DIN[17]	—	—	Y[7]	—	Cb [1]	Y[5]
DIN[18]	—	—	Y[8]	—	Cb [2]	Y[6]
DIN[19]	—	—	Y[9]	—	Cb [3]	Y[7]
DIN[20]	—	—	—	—	Cb [4]	Y[8]
DIN[21]	—	—	—	—	Cb [5]	Y[9]
DIN[22]	—	—	—	—	Cb [6]	Y[10]
DIN[23]	—	—	—	—	Cb [7]	Y[11]

ADI Confidential for Newvision under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

Table 58. UDP8/Generic8 Color Map

19.3.2.1.1 **Input Bits**	**UDP8 (8bit)**	**UDP8 (8bit)**	**UDP8 (8bit) DBL**	**Generic8 (8bit)**	**Generic8 (8bit)**	**Generic8 (8bit) DBL**
DIN[0]	P[0]	P[0]	P[0]	P[0]	P[0]	P[0]
DIN[1]	P[1]	P[1]	P[1]	P[1]	P[1]	P[1]
DIN[2]	P[2]	P[2]	P[2]	P[2]	P[2]	P[2]
DIN[3]	P[3]	P[3]	P[3]	P[3]	P[3]	P[3]
DIN[4]	P[4]	P[4]	P[4]	P[4]	P[4]	P[4]
DIN[5]	P[5]	P[5]	P[5]	P[5]	P[5]	P[5]
DIN[6]	P[6]	P[6]	P[6]	P[6]	P[6]	P[6]
DIN[7]	P[7]	P[7]	P[7]	P[7]	P[7]	P[7]
DIN[8]	P[0]	—	P[0]	P[0]	—	P[0]
DIN[9]	P[1]	—	P[1]	P[1]	—	P[1]
DIN[10]	P[2]	—	P[2]	P[2]	—	P[2]
DIN[11]	P[3]	—	P[3]	P[3]	—	P[3]
DIN[12]	P[4]	—	P[4]	P[4]	—	P[4]
DIN[13]	P[5]	—	P[5]	P[5]	—	P[5]
DIN[14]	P[6]	—	P[6]	P[6]	—	P[6]
DIN[15]	P[7]	—	P[7]	P[7]	—	P[7]
DIN[16]	P[0]	—	—	P[0]	—	—
DIN[17]	P[1]	—	—	P[1]	—	—
DIN[18]	P[2]	—	—	P[2]	—	—
DIN[19]	P[3]	—	—	P[3]	—	—
DIN[20]	P[4]	—	—	P[4]	—	—
DIN[21]	P[5]	—	—	P[5]	—	—
DIN[22]	P[6]	—	—	P[6]	—	—
DIN[23]	P[7]	—	—	P[7]	—	—

Time 0
Time 1
Time 2

ADI Confidential for New Vision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

19.4 Status and Debug Registers

If the video crossbar is not functioning properly, perform the following troubleshooting process:

- Verify the physical connections in the system and check that the voltage levels on the input pins are within the expected range.
- Confirm that the video input is started and that the video stream is being transmitted.
- Verify that the GMSL link is locked and that the control channel (I2C/UART) is accessible.
- Ensure that the video stream is routed to the correct GMSL PHY using the TX_SPLT_MASK_A/B registers of the relevant video pipe.
- Verify that the video is locked on the deserializer side of the link.
- Ensure that the video input format is stable.
- Confirm that the video crossbar has been correctly configured according to the desired output format.

19.5 Feature Availability by Device Family

Video crossbar is available in all GMSL2 devices except eDP/DP deserializers (Table 59).

Table 59. Video Crossbar Support by Device Family

Device Family	Video Crossbar
HDMI Serializers	X
Advanced HDMI Serializers	X
DSI Serializers	X
Advanced DSI Serializers	X
CSI-2 Serializers	X
Advanced CSI-2 Serializers	X
CSI-2 Camera Deserializers	X
CSI-2 Display Deserializers	X
OLDI Deserializers	X
eDP/DP Deserializers	
CSI-2 Quad Deserializers	X

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

20 Color Lookup Table (LUT)

20.1 Overview

All GMSL2 OLDI and eDP/DP deserializers contain Color Lookup Tables (LUT) that support automatic 1:1 translation of all RGB888 pixel color values to any desired RGB888 pixel color values. The LUTs are generally used for color filtering or gamma correction.

The color LUT enables 1:1 translation of 8-bit color input data to any 8-bit color output values. The color LUT comprises three discrete color channels that accept 8-bit wide input for each color (i.e., red, green, and blue) with 256-entry depth. Each LUT RGB color sub-system has a separate configuration register block that can be programmed to automatically translate incoming pixel data. If color translation is needed, all three color LUTs must be configured and enabled.

No configuration of the color LUTs is required for a straight 1:1 mapping of the input data to output data. By default, the color LUTs are disabled and bypassed such that pixel values are not translated to other values. If pixel value translation is needed, the LUT registers should be programmed before enabling the color LUT block.

All OLDI deserializers have one video pipe and one LUT, while eDP/DP deserializers have four video pipes and four LUTs.

Figure 49 shows the LUT and LUT RGB sub-system architecture with incoming pixel data, the RBG LUTs, and the output pixel data. Note that each color can be user-programmed independently.

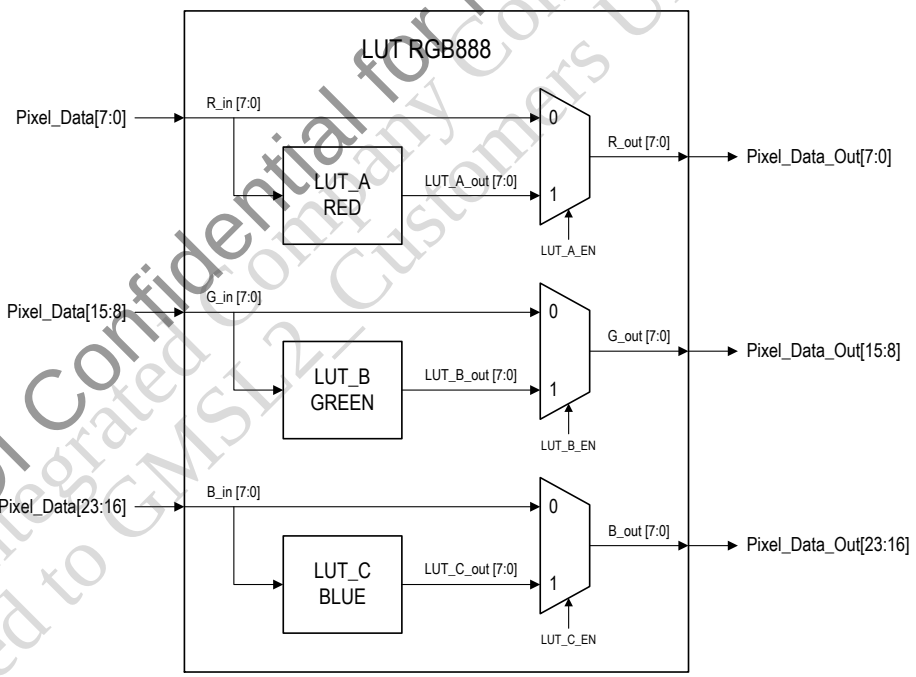


Figure 49. RGB Pixel Data Translation using Color Lookup Tables (LUT)

ADI Confidential for New Vision Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

20.2 Configuration

At power-up, the LUTs are assigned random values but are not enabled. The LUTs should first be configured via the appropriate register space(s) before setting the register enable bit.

The beginning of the register address space for each LUT corresponds to the LSB of the respective color.

The LUT can be reverted to a 1:1 straight color mapping after the random value assignment at power-up. For example, use the following programming pattern to revert the red color LUT of the OLDI deserializer to 1:1 mapping values:

- 0x1000 = 0x00
- 0x1001 = 0x01
- 0x1002 = 0x02
- …
- 0x10FF = 0xFF

Enable the LUT using the LUT_x_EN register bit after programming the LUT registers.

The tables below contain the LUT register information. These include the address range for each LUT and their corresponding enable bits. After configuring, the incoming pixel data will automatically translate to the new output LUT values after enabling the LUT enable bit.

Note: LUTs cannot be partially enabled. If color translation is needed, all three LUTs must be enabled. See *Enabling LUTs* for more information.

20.2.1 OLDI Deserializers

Table 60. OLDI Deserializer LUT Registers

Field / Register Name	Address Range / Register Name	Bit Position	Default Value	Description
LUT_A (RED) register space	0x1000 – 0x10FF		Random	Address Range for RED LUT Data
LUT_B (GREEN) register space	0x1100 – 0x11FF		Random	Address Range for GREEN LUT Data
LUT_C (BLUE) register space	0x1200 – 0x12FF		Random	Address Range for BLUE LUT Data
LUT_A_EN	0x01CD / OLDI0	0	0: Disabled	Enables LUT_A (RED) translation
LUT_B_EN	0x01CD / OLDI0	1	0: Disabled	Enables LUT_B (GREEN) translation
LUT_C_EN	0x01CD / OLDI0	2	0: Disabled	Enables LUT_C (BLUE) translation

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

20.2.2 eDP/DP Deserializers

Table 61. eDP/DP Deserializer LUT Registers

Pipe	Field / Register Name	Register Address	Bit Pos.	Default Value	Description
X	LUT_A_X (RED) reg. space	0x2000 – 0x20FF		Random	Address Range for Pipe-X RED LUT Data
	LUT_B_X (GREEN) reg. space	0x2100 – 0x21FF			Address Range for Pipe-X GREEN LUT Data
	LUT_C_X (BLUE) reg. space	0x2200 – 0x22FF			Address Range for Pipe-X BLUE LUT Data
	LUT_A_EN_X	0x01DD DISP_CTRL_X	0	0: Disabled	Enables Pipe-X LUT_A (RED) translation
	LUT_B_EN_X		1		Enables Pipe-X LUT_B (GREEN) translation
	LUT_C_EN_X		2		Enables Pipe-X LUT_C (BLUE) translation
Y	LUT_A_Y (RED) reg. space	0x2400 – 0x24FF		Random	Address Range for Pipe-Y RED LUT Data
	LUT_B_Y (GREEN) reg. space	0x2500 – 0x25FF			Address Range for Pipe-Y GREEN LUT Data
	LUT_C_Y (BLUE) reg. space	0x2600 – 0x26FF			Address Range for Pipe-Y BLUE LUT Data
	LUT_A_EN_Y	0x01FD DISP_CTRL_Y	0	0: Disabled	Enables Pipe-Y LUT_A (RED) translation
	LUT_B_EN_Y		1		Enables Pipe-Y LUT_B (GREEN) translation
	LUT_C_EN_Y		2		Enables Pipe-Y LUT_C (BLUE) translation
Z	LUT_A_Z (RED) reg. space	0x2800 – 0x28FF		Random	Address Range for Pipe-Z RED LUT Data
	LUT_B_Z (GREEN) reg. space	0x2900 – 0x29FF			Address Range for Pipe-X GREEN LUT Data
	LUT_C_Z (BLUE) reg. space	0x2A00 – 0x2AFF			Address Range for Pipe-Z BLUE LUT Data
	LUT_A_EN_Z	0x021D DISP_CTRL_Z	0	0: Disabled	Enables Pipe-Z LUT_A (RED) translation
	LUT_B_EN_Z		1		Enables Pipe-Z LUT_B (GREEN) translation
	LUT_C_EN_Z		2		Enables Pipe-Z LUT_C (BLUE) translation
U	LUT_A_U (RED) reg. space	0x2C00 – 0x2CFF		Random	Address Range for Pipe-U RED LUT Data
	LUT_B_U (GREEN) reg. space	0x2D00 – 0x2DFF			Address Range for Pipe-U GREEN LUT Data
	LUT_C_U (BLUE) reg. space	0x2E00 – 0x2EFF			Address Range for Pipe-U BLUE LUT Data
	LUT_A_EN_U	0x023D DISP_CTRL_U	0	0: Disabled	Enables Pipe-U LUT_A (RED) translation
	LUT_B_EN_U		1		Enables Pipe-U LUT_B (GREEN) translation
	LUT_C_EN_U		2		Enables Pipe-U LUT_C (BLUE) translation

Note: In typical single-video applications with the eDP/DP deserializers, only pipe X contains video, so only this LUT needs to be programmed.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

20.2.3 Readback Restriction

The following sequence is common in typical use cases:

- Program the LUT translation values.
- Readback the LUT translation values (to ensure proper programming).
- Enable the LUT.

LUT translation values will not readback correctly (e.g., 0xFF) in GMSL2 devices unless valid video data is received by the deserializer. The internal block is powered down until valid video data is received.

However, writes to the LUT are always valid (even when valid video data does not exist) if the I2C/UART ACK is received to confirm the register write. To confirm the LUT values by reading back via I2C/UART, valid video data is required.

20.2.4 Enabling LUTs

Although LUT_A, LUT_B, and LUT_C have separate enable registers in GMSL2 devices, the LUTs cannot be enabled partially. All LUTs (i.e., LUT_A, LUT_B, and LUT_C) in a single video pipe must be enabled/disabled together for proper operation.

If configuration of a single color LUT is required, but the others are not, configure the single LUT with the desired translation values and the remaining LUTs with 1:1 mapping values. For example, if only red color translation is needed, populate LUT_A with the desired translation values and fill green (LUT_B) and blue (LUT_C) with the linear 1:1 translation values.

20.3 Feature Availability by Device Family

Color Lookup Tables (LUT) are available on all GMSL2 OLDI and eDP/DP deserializers (Table 62).

Table 62. Color Lookup Table Support by Device Family

Device Family	Color Lookup Tables (LUT)
HDMI Serializers	
Advanced HDMI Serializers	
DSI Serializers	
Advanced DSI Serializers	
CSI-2 Serializers	
Advanced CSI-2 Serializers	
CSI-2 Camera Deserializers	
CSI-2 Display Deserializers	
OLDI Deserializers	X
eDP/DP Deserializers	X
CSI-2 Quad Deserializers	

Maxim Integrated Confidential, Distribution Limited to GMSL2 Customers Under NDA
Provided for New Design Only
ADI Confidential

21 Frame Sync

21.1 Overview

Certain GMSL applications may require the aggregation of multiple video sources/inputs into a single MIPI output. The packet-based MIPI protocol supports video aggregation via merging separate streams on a first come, first serve (FCFS basis). The SoC can process the merged stream by allocating the constituent streams to different virtual channels. If the application requires that the video streams be time-synchronized for advanced processing, frame sync signals (FSYNC) are used to enable video stream synchronization and alignment. Here, the SoC sends FSYNC signals to synchronize video frames at the video source(s). FSYNC signals allow video frames to be processed simultaneously and provide the option to concatenate multiple video streams into a single synchronized video frame.

21.2 Operation

Compatible GMSL2 CSI-2 deserializers (reference Feature Availability by Device Family) feature an internal frame sync (FSYNC) generator block and/or pin that accepts external FSYNC generation (e.g., from the SoC) with low-latency GPI-to-GPO tunneling. These devices interface with images sensors with an external FSYNC input.

The FSYNC signal is generated periodically or pseudo-periodically. The frame length is defined by the number of pixels in each frame or by referencing the 25MHz crystal clock (eliminating dependence on the image sensor PCLK). After generation, the FSYNC signal is transmitted via the reverse channel on the serial link to connected serializers and, subsequently, image sensors to synchronize the captured video frames.

Video frame synchronization occurs by synchronizing the vertical sync (VS) signals of the various video streams at the image sensors (Figure 50 and Figure 51). This initial synchronization prepares the video streams for processing by the SoC when output from the serial link system.

Frame synchronization permits the deserializer to concatenate the received video streams. With concatenation, the deserializer can output video frames in line-concatenated (4WxH) or line-interleaved (Wx4H) formats.

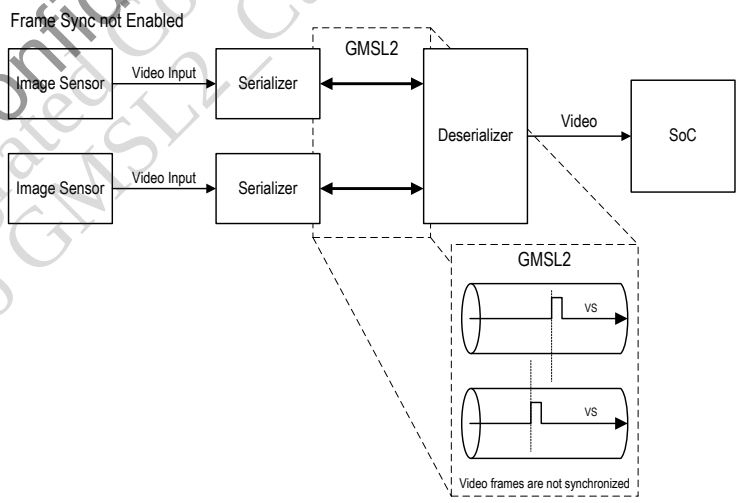


Figure 50. VS Signal Path (Frame Sync not Enabled)

ADI Confidential for Navision under NDA
Analog Devices Confidential Property
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

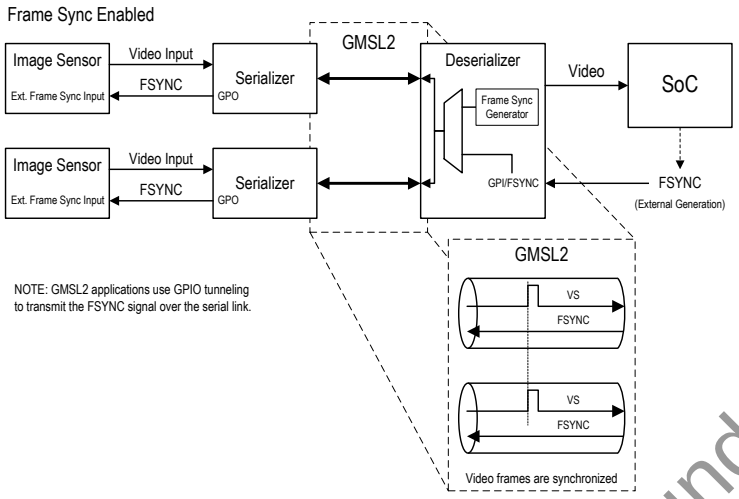


Figure 51. VS Signal Path (Frame Sync Enabled)

Frame sync is typically used in serial link systems featuring a single deserializer connected to multiple serializers; in extended systems with multiple image sensor arrays, two or more deserializers may be required. Here, the frame sync signal must be coordinated between multiple deserializers and all connected image sensors. One deserializer will be configured as a master for frame sync generation and the other(s) will be configured as frame sync slave(s). After the master deserializer generates the frame sync signal, it is routed to the slave deserializer(s); all deserializers send the frame sync signal to the connected serializers and image sensors (as described above).

Note that the frame sync signal pattern differs in GMSL1 mode. In GMSL2 mode, (slave) deserializers transmit the frame sync signal as a standard GPIO signal via GPI-GPO tunnel(s). In GMSL1 mode, there is only one shared GPI-GPO tunnel that is used exclusively by the main Control Channel or for GPI-GPO tunneling (or frame sync) at a time. When a frame sync signal is transmitted in GMSL1 mode, I2C is temporarily paused by clock stretching. In GMSL2 mode, there are no restrictions on frame sync transmissions nor any adverse effects on other channels.

ADI Confidential for New Version under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

21.3 Configuration

Frame sync applications require that all devices in the serial link system be compatible with frame sync. For compatible systems, the GMSL2 deserializer must be configured before frame sync can be enabled.

1. Determine the Frame Sync Mode required for the application and program the FSYNC_MODE register (Table 63).
2. Configure the Frame Sync Method (FSYNC_METH) to set the method FSYNC signal generation (Table 66).
3. GMSL2 Devices: Determine and configure the GPIO ID in both the deserializer and serializer.

The connected image sensor(s) must be able to accept external FSYNC signals. Configure the image sensor before proceeding with GMSL2 device configuration.

21.3.1 Frame Sync Mode

The topology of the serial link system determines the frame sync mode of operation that should be programmed. There are four possible modes selected by the FSYNC_MODE register in the deserializer (Table 63).

Table 63. Frame Sync Mode (FSYNC_MODE)

Value	Decode
00	Frame sync generation is on (internal frame sync mode). GPIO is not used as FSYNC input or output.
01	Frame sync generation is on (internal frame sync mode). GPIO is used as FSYNC output and drives a slave device.
10	Frame sync generation is off. In GMSL1 mode, GPIO is used as FSYNC input driven by a master device.
11	Frame sync generation is off. In GMSL1 mode, GPIO is not used as FSYNC input or output.

A typical frame sync application is in a serial link system comprising multiple serializers connected to one deserializer. In this scenario, the deserializer can either generate the FSYNC signal (enabled by programming register FSYNC_MODE = 00) or receive the FSYNC signal from an external source (FSYNC_MODE = 10). Any of the FSYNC methods may be used, depending on application (refer to Frame Sync Method for more information).

If multiple deserializers are used in the system, one can be configured as a frame sync master for the other connected slave deserializers. Program register FSYNC_MODE = 01 in the master deserializer to output the FSYNC signal from an MFP pin; then, program register FSYNC_MODE = 10 in the slave deserializer(s) to receive the FSYNC signal on a GPI.

If frame sync is not required, set FSYNC_MODE = 11 in the device.

The frame sync applications are similar for both GMSL2 and GMSL1 modes, however, there are differences. For details, reference Table 64 for GMSL2 applications and Table 65 for GMSL1 applications. The FSYNC_METH register is explained below in the Frame Sync Method section.

Note: The tables below use the MAX9296A/B as an example device. Refer to device datasheets for specific register details.

ADI Confidential for New Revision under NDA
Analog Devices Confidential, Subject to Change without Notice
Maxim Integrated Confidential, Company Confidential,
Provided to GMSL Customers Under NDA

Table 64. Frame Sync Modes and Methods (GMSL2)

FSYNC_MODE	FSYNC_METH	GMSL2 Mode Explanation	GMSL2 Usage
00	Manual; Auto; Semi-Auto	Frame sync generation is ON. The deserializer GPIO is NOT used as an output for frame sync	1. Internal frame sync 2. No sync signal output
01	Manual; Auto; Semi-Auto	Frame sync generation is ON. The deserializer GPIO used as an output for frame sync	1. Internal frame sync 2. Sync signal output at FRSYNC_OUT (MFP0 for MAX9296)
10	N/A	Frame sync generation is off. GPIO is not used as FSYNC output	1. No internal frame sync 2. May use GPIO forwarding for external frame sync
11	N/A	Frame sync generation is off. GPIO is not used as FSYNC output	

Note: Quad deserializer registers may differ than those in *Table 64*. Refer to the quad deserializer frame synchronization section for additional details.

Table 65. Frame Sync Modes and Methods (GMSL1)

FSYNC_MODE	FSYNC_METH	GMSL1 Mode Explanation	GMSL1 Usage
00	Manual; Auto; Semi-Auto	Frame sync generation is ON. The frame sync pin is in HIGH-Z mode	1. Internal frame sync 2. No sync signal output
01	Manual; Auto; Semi-Auto	Frame sync generation is ON. The deserializer GPIO is NOT used as an output for frame sync	1. Internal frame sync 2. Sync signal output at FRSYNC_OUT (MFP0 for MAX9296)
10	N/A	Frame sync generation is off. FSYNC pin is driven by a master device	1. No internal frame sync 2. Device is configured as a slave device to receive frame sync signal from external master device 3. The frame sync signal input from FRSYNC_IN, which is controlled by FSYNC_OUT_PIN
11	N/A	Frame sync generation is off. GPIO is not used as FSYNC output	1. No internal frame sync 2. GPI-to-GPO tunnel can be used to transmit external frame sync signal to serializer 3. Use GPI_SEL to choose GPI_x; GPI_COMP_EN needs to be turned on if PKTCC_EN is on

In GMSL1 mode, FSYNC_MODE = 10 may be used to configure the deserializer in slave mode to receive the frame sync signal from another deserializer (configured as a master: FSYNC_MODE = 01).

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_ Customers Under NDA

21.3.2 Frame Sync Method

When the frame sync generator is enabled, the FSYNC_METH register is used to set the method of frame sync generation. The frame sync generator has three methods of operation: Manual, Auto, and Semi-Auto. When the deserializer receives an external FSYNC signal, the FSYNC_METH register setting has no effect.

Table 66. FSYNC Methods of Operation (FSYNC_METH)

FSYNC_METH	Decode
0b00	Manual: User-defined clock periods for FSYNC generation.
0b01	Semi-Auto: FSYNC period is automatically set; use this method when only the row counter is reset.
0b10	Auto: FSYNC period is automatically set; use this method when both the row and column counters are reset.
0b11	Reserved

21.3.2.1 Overlap Window

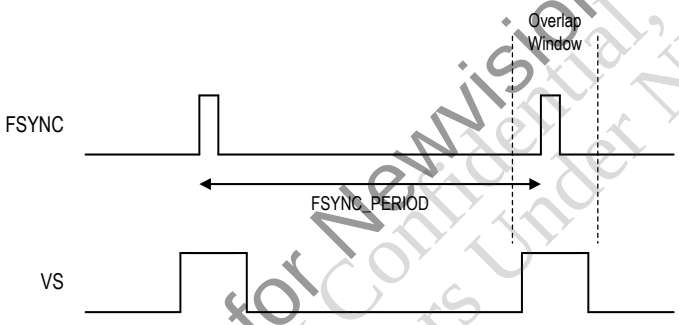


Figure 52. Overlap Window (FSYNC and VS signals)

The overlap window defines a threshold that starts from the VS rising edge of the fastest image sensor and ends when a FSYNC signal is received (Figure 52). It is measure in terms of PCLK cycles with the OVLP_WINDOW_L and OVLP_WINDOW_H bitfields. The window is used to check that the FSYNC signal is successfully sent from the deserializer to the image sensor and the corresponding VS signal is successfully sent from the image sensor to the deserializer. The overlap window should be programmed according to calculations considering the maximum frame synchronization skew and the maximum frequency difference (ppm) of all camera links.

Note: The overlap window may be disabled if there is an ISP (Image Signal Processor) between the image sensor and the serializer. The ISP may delay frame generation and cause a failure in the overlap protection.

21.3.2.2 Manual Frame Synchronization

Manual mode is the simplest (and most common) frame sync method. This method uses fixed clock periods defined by FSYNC_PERIOD register for frame sync signal generation. The FS_USE_XTAL register is used to specify whether the PCLK or the crystal oscillator (GMSL2 mode only) is used as the clock source. See the Crystal Clock Option section below for additional information.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL Customers Under NDA

21.3.2.2.1 Crystal Clock Option

GMSL2 devices have an integrated crystal clock source that can be used as an alternative to the PCLK for frame sync generation. In manual mode, a 25MHz clock from the crystal oscillator can be used to generate the frame sync signal. Set FS_USE_XTAL=1 to periodically generate the frame sync signal using the 25MHz clock as the base period. Set FS_USE_XTAL=0 if using the PCLK.

21.3.2.3 Automatic Frame Synchronization

Automatic mode is recommended for use in systems with image sensors that run in continuous mode and reset both row and column counters (in the image sensor) upon receipt of the frame sync signal (e.g., OV10635). The FSYNC_PERIOD register is not user-configured in this mode.

When automatic frame synchronization is enabled, the deserializer detects and analyzes free-running VS signals on the active GMSL links. After determining the slowest VS pulse and allowing for timing margin adjustments, the deserializer sends the FSYNC signal on the reverse channel. Frame synchronization is locked when all received VS signals are synchronized for two consecutive cycles.

1. Automatic frame synchronization requires that the initial VS signals of each link be free running. The master link frame length (FL) is counted in terms of PCLK cycles until the deserializer detects VS signals from all connected and enabled GMSL links.
    - Note: The master link is either the first link that locks or defined by the user.
2. The first FSYNC signal is sent on the reverse channel. The row and column counters of each connected image sensor are reset to align the start of each link's subsequent frame.
3. The aligned VS signals are received by the deserializer. These signals will arrive at slightly different times due to differences on each link (i.e., clock frequency offset and transmission timing skew). The difference between the master link and the slowest link is calculated in terms of $master\ pixel\ clock - t_{DIFF}$.
4. The deserializer dynamically adjusts each link's FSYNC signal generation to compensate for link differences and ensure that the frames arrive at the deserializer synchronized.
    - Loss of lock will occur if the difference between the arrival of the earlier and later signal exceeds the threshold.

ADI Confidential for new version under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers under NDA

21.3.2.4 Semi-Automatic Frame Synchronization

Semi-automatic mode is recommended for use in systems with image sensors that reset only the row counter upon receipt of the frame sync signal (e.g., OV10640).

This mode is similar to *Automatic Frame Synchronization*. However, in semi-automatic mode, the deserializer does not check the skew between the different GMSL links (t_{DIFF}); adjustments of timing margins are instead measured from the master link. The FSYNC signal resets only the row counter in the image sensors.

1. Semi-automatic frame synchronization requires that the initial VS signals of each link be free running. The master link frame length (FL) is counted in terms of PCLK cycles until the deserializer detects VS signals from all connected and enabled GMSL links.
    a. The master link is either the first link that locks or defined by the user.
2. The first FSYNC signal is sent on the reverse channel to align the VS signals of connected image sensors. This reset causes the subsequent frame starts of the connected image sensors to be within two rows of each other.
3. The aligned VS signals are received by the deserializer. Note that these signals will arrive at slightly different times due to differences on each link (i.e., clock frequency offset and transmission timing skew) and that only the rows (and not the columns) are reset.
4. The deserializer dynamically adjusts FSYNC signal generation based on timing measurements of the master link to ensure that the frames arrive at the deserializer synchronized.
    a. Loss of lock will occur if the difference between the arrival of the earlier and later signal exceeds the threshold.

21.3.3 Status and Error Bits

The frame sync locking mechanism and associated registers are the same for all three frame synchronization methods. The frame sync lock bit (FSYNC_LOCKED) indicates that frame sync has successfully locked. FSYNC_LOSS_OF_LOCK goes high if frame sync lock is lost. Other important registers for the locking mechanism include the FSYNC_LOSS_OF_LOCK, FRM_DIFF_ERR_THR, FSYNC_ERR_CNT, and FSYNC_ERR_THR register bits.

1. CALC_FRM_LEN: The calculated VS period (number of PCLKs) of master link in automatic or semi-automatic synchronization mode.
2. FRM_DIFF: The difference between the fastest and the slowest frame in terms of master PCLK cycles
3. FRM_DIFF_ERR_THR: Error threshold for difference between the earliest and latest VS signals in terms of PCLK cycles. The default is 40µs for a 96MHz PCLK. The function is disabled when all 13 bits are 0.
4. FSYNC_ERR_CNT: Frame sync error counter. Incremented when error condition is detected and flagged. Resets to 0 when read or when FSYNC_LOCKED goes high.
5. FSYNC_ERR_THR: Frame sync error reporting threshold. FSYNC_ERR_FLAG is asserted when FSYNC_ERR_CNT is greater than FSYNC_ERR_THR.

The frame sync error status (FSYNC_ERR_FLAG) can be configured to drive the ERRB pin (FSYNC_ERR_OEN = 1).

The deserializer checks that enabled links have valid VS signals and that the VS signals occur within the overlap window. The overlap window is programmed via the OVLP_WINDOW bits. The overlap window may be disabled if there is an ISP (Image Signal Processor) between the image sensor and the serializer. The ISP may delay frame generation and cause a failure in the overlap protection.

Analog Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA
ADI Confidential for New Version under NDA

21.3.4 Configuration Process

21.3.4.1 GMSL2 Deserializer in GMSL2 Mode

The following process is used to enable frame sync for a GMSL2 deserializer in GMSL2 mode:

- Write AUTO_FS_LINKS high (default) to enable FSYNC on all enabled links. Alternatively, the bits FS_EN_X/Y/Z/U can be used to enable FSYNC on video pipes individually.
- Disable all unused video controllers in the deserializer. Unused video controllers can be identified by the absence of VIDEO_LOCK. For example, if VIDEO_LOCK is only indicated for controllers “X” and “Y”, then controllers “Z” and “U” must be disabled. Video controller configuration registers are in REG2.
- Choose the proper master link for the frame sync generation via FSYNC_2. This can be any video controller with VIDEO_LOCK.
    - Note: If a master link is not designated, the frame sync block will default the master link designation to the first link to achieve link lock.
- Configure the GPIO TX ID for frame sync so that the FSYNC pulse will be output to the correct GPIO pin of the serializer. This is configured in the FSYNC_17 register.
- Set up the GPIO RX ID for the GPIO pin on the serializer that is to receive the FSYNC signal. Ensure that the deserializer output pin and the serializer input pin are correctly mapped. Reference the General-Purpose Input and Output (GPIO) section for more information.
- Set the Frame Sync Mode and Frame Sync Method (see above). Relevant registers are in FSYNC_0.
- If necessary, set the overlap window (OVLP_WINDOW).
    - OVLP_WINDOW: Defines a threshold which starts from the VS rising edge of fastest image sensor and ends when a frame sync signal is seen. Measured in pixel clocks cycles.
        - OVLP_WINDOW_L in register FSYNC_10
        - OVLP_WINDOH_H in register FSYNC_11
    - Note that OVLP_WINDOW may be disabled if there is an ISP (Image Signal Processor) between the image sensor and the serializer. The ISP may delay the frame generation and the overlap protection will fail.

21.3.4.2 GMSL2 Deserializer in GMSL1 Mode

GMSL2 deserializers configured for backward compatible GMSL1 mode can use frame sync. Operation in GMSL1 mode is similar to operation GMSL2 mode (see Table 65 above). Please note:

1. Link A is in Pipe X and Link B is in Pipe Y.
2. For the MAX9296A/B, Link A and Link B have discrete GMSL1 blocks containing separate GMSL1 registers for each.
3. The EN_FSYNC_TX register must be set to use the frame sync signal.
4. Register GPI_EN is used to enable GPI-to-GPO transmission (default is on). Ensure that the MFP is available for GPI usage.
5. GPI_COMP_EN must be enabled if PCKTCC_EN is on and GPI-to-GPO is used.

ADI Confidential for New Vision under NDA
Analog Devices Confidential
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

21.3.5 Example Script Code

21.3.5.1 MAX9296A/B – Manual Frame Sync

```
# Frame sync setup
#
# Disable unused controllers in the deserializer. In this case, all controllers are being
used.
0x90,0x02,0xF3
#Manually select the master link
0x90,0x3E2,0x20
# Disable overlap window
0x90,0x3EA,0x00
0x90,0x3EB,0x00
# Set 30Hz signal manual FSYNC generation. Count PCLKs on Master Link before sending FSYNC
pulse.
0x90,0x3E6,0xC3
0x90,0x3E7,0x5D
# GPIO ID setup to output FSYNC on CFG0 (MFP1) on MAX9295A and on MFP4 of MAX9296B
0x90,0x3F1,0x08
# Set the FSYNC method to MANUAL, the FSYNC mode to generate an FSYNC signal in the
deserializer, and to output from MFP4.
0x90,0x3E0,0x04
# Set MFP1 of each MAX9295A to receive data from the MAX9296B
0x80,0x2C1,0x84
0x40,0x2C1,0x84
#
# End of Frame Sync Setup
```

21.3.5.2 MAX9296A/B GMSL1 Mode – Automatic Frame Sync

```
//FSYNC
//turn on Auto, FSYNC_MODE = 0b01
0x04,0x91,0x03,0xE0,0x06,0xF,
//turn off the overlap window
0x04,0x90,0x03,0xEA,0x00,
0x04,0x91,0x03,0xEB,0x00,0x1F,
//enable pipe0 and pipe 1
//using FS_GPIO_TYPE = 0, gmsl_1 GPIO
0x04,0x91,0x03,0xEF,0x03,0x1F,
//turn on EN_FSYNC_TX
0x04,0x91,0x0B,0x08,0x10,0x10,
0x04,0x91,0x0C,0x08,0x10,0x10,
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute Only to NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

21.4 Feature Availability by Device Family

Frame sync is available on select GMSL2 deserializers. See the table below for part numbers with frame sync capabilities (Table 67).

Note: All GMSL1 and GMSL2 camera serializers are compatible with FSYNC serial link applications.

Table 67. Frame Sync Support by Part Number

Device Family	Frame Sync
HDMI Serializers	
Advanced HDMI Serializers	
DSI Serializers	
Advanced DSI Serializers	
CSI-2 Serializers	
Advanced CSI-2 Serializers	
CSI-2 Camera Deserializers	MAX96784 MAX9296A MAX9296B
CSI-2 Display Deserializers	MAX96756 MAX96756F MAX96758F MAX96758
OLDI Deserializers	
eDP/DP Deserializers	
CSI-2 Quad Deserializers	MAX96712 MAX96722

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

22 Video Sync Pulse Outputs

22.1 Overview

GMSL2 serializers and deserializers can output video sync signals (i.e., VS, HS, and DE) to aid in system-level debugging. The polarity of these signals can be flipped to accommodate requirements of video formatting and connected devices. The sync signal timing can also be modified with the video timing generator (VTG). Reference the Video Timing Generator (VTG) and Video Pattern Generator (VPG) section for additional information.

22.2 Operation

Videos may use either active-high or active-low VS and HS signals. The polarity of HS, VS, and DE signals are individually configurable with GMSL2 devices. This provides flexibility in system design and ensures compatibility with different cameras and displays.

In serializers, the sync signal polarity can be inverted using the following bits in the VTX register block:

- VS_INV – Inverts the VS pulse.
- HS_INV – Inverts the HS pulse.
- DE_INV – Inverts the DE pulse.

In deserializers, the sync signal polarity can be inverted using the following bits in the VRX register block:

- CROSS_VS_I – Inverts the VS pulse.
- CROSS_HS_I – Inverts the HS pulse.
- CROSS_DE_I – Inverts the DE pulse.

22.3 Configuration

HS, VS, or DE signals can be routed to any available MFP/GPIO pin. To configure sync signal MFP/GPIO output, first ensure that the alternate functions of MFP(s) are disabled and GPIO output is enabled. For a particular MFP/GPIO, set the following:

- GPIO_RX_EN = 0
- GPIO_TX_EN = 0
- GPIO_OUT_DIS = 0
- PULL_UPDN_SEL = 00
- OUT_TYPE = 1

The pins and register configuration used for sync signal output vary by device family. See the following tables (Table 68 – Table 75) for details for each device family.

Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Vision under NDA

Table 68. HDMI Serializers and Advanced HDMI Serializers

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	Yes	GPIO17	Register: VTX1 Bit: VS_OUT_EN
HS	Yes	GPIO16	Register: VTX1 Bit: HS_OUT_EN
DE	No	N/A	N/A

Table 69. DSI Serializers, Advanced DSI Serializers, and Advanced CSI-2 Serializers

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	No	N/A	N/A
HS	No	N/A	N/A
DE	No	N/A	N/A

Table 70. CSI-2 Serializers

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS, HS, DE from CSI-2 Input	No	N/A	N/A
VS from REF_VTG block	Yes	Selectable	Use REF_VTG block to set the sync high and low period Set REFGEN_EN=1 Set GEN_VS=1 Select which GPIO to output with VS_GPIO[4:0] Set VSEN = 1 to enable the output
HS from REF_VTG block	Yes	Selectable	Use REF_VTG block to set the sync high and low period Set REFGEN_EN=1 Set GEN_HS=1 Select which GPIO to output with HS_GPIO[4:0] Set HSEN = 1 to enable the output
PCLK from REF_VTG block	Yes	Selectable	Set REFGEN_EN=1 Select predefined frequency in PLL (REFGEN_PREDEF_FREQ [1:0]) or RCLK frequency (RCLKEN_Y) = 1 for desired PCLK frequency Select which GPIO to output with PCLK_GPIO[4:0] Set PCLKEN = 1 to enable the output

Note: If PCLK, HS, and VS signals are mapped to the same pin, the priority from high to low is HS, VS, and PCLK. Refer to the *CSI-2 Serializer Sync Pulse Configuration* section below for setting high and low periods for sync signals.

ADI Confidential for Television under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customer Under NDA

Table 71. CSI-2 Camera Deserializers (except the MAX96784)

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	Yes	MFP3	VS_OUT1[2] enables VS output to MFP3 VS_OUT1[1:0] selects which video pipe to output
VS	Yes	MFP0	VS_OUT2[2] enables VS output to MFP0 VS_OUT2[1:0] selects which video pipe to output
HS	No	No	N/A
DE	Yes	MFP0/MFP3	HS_OUT1[2] enables DE output to MFP0 and MFP3 HS_OUT1[1:0] selects which video pipe to output

Note: If DE and VS signals are mapped to the same pin, VS signal has higher priority.

Table 72. CSI-2 Display Deserializers and the MAX96784

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	Yes	MFP2	VS_OUT1[2] enables VS output to MFP2 VS_OUT1[1:0] selects which video pipe to output VS_OUT2[2:0] is reserved
HS	No	N/A	N/A
DE	Yes	MFP3	HS_OUT1[2] enables DE output to MFP3 HS_OUT1[1:0] selects which video pipe to output

Table 73. OLDI Deserializers

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	Yes	GPIO1	Register: OLDI2 Bit: VS_OUT_EN
HS	Yes	GPIO1	Can be output to GPIO1 by swapping HS and VS with the video crossbar. Not available at the same time as VS output. Set CROSS25[4:0] = 0x18 and CROSS24[4:0] = 0x19
DE	No	N/A	Can be output to GPIO1 by swapping DE and VS with the video crossbar. Not available at the same time as VS output. Set CROSS25[4:0] = 0x1A and CROSS26[4:0] = 0x19

Note: The register HS_OUT_EN is reserved.

ADI Confidential for NEW VISION under NDA
Analog Devices Confidential Distribution only to Maxim Integrated Company Confidential Customers Under NDA
Provided to GMSL2_Customers

Table 74. eDP/DP Deserializers

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	Yes	GPIO1	Register: VRX__X:VIDEO_CTRL_OUT Bit: VS_OUT_EN (also available on Pipe Y, Z, and U in their relevant VRX register)
HS	Yes	GPIO0	Register: VRX__X:VIDEO_CTRL_OUT Bit: HS_OUT_EN (also available on Pipe Y, Z, and U in their relevant VRX register)
DE	No	N/A	N/A

Table 75. CSI-2 Quad Deserializers

Sync Signal	Sync Output Available	Sync Output Pin	Sync Output Configuration
VS	Yes	MFP4	Register: HVD_GPIO_CTRL Bit: VS_EN enables VS output to MFP4 Bit: HVD_SEL[2:0] selects which video pipe to output
HS	Yes	MFP13	Register: HVD_GPIO_CTRL Bit: HS_EN enables HS output to MFP13 Bit: HVD_SEL[2:0] selects which video pipe to output
DE	Yes	MFP14	Register: HVD_GPIO_CTRL Bit: DE_EN enables DE output to MFP14 Bit: HVD_SEL[2:0] selects which video pipe to output

Note: Set register the DIS_LOC_CC_P2 = 1 to output DE and VS. HS output functionality requires either that the connected serializer supports HS (e.g., parallel input) or that the HS signal is internally generated in the serializer (e.g., pattern generation).

ADI Confidential for new version under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

22.3.1 CSI-2 Serializer Sync Pulse Configuration

CSI-2 Serializers can use the REF_VTG block to output sync signals from MFP pins. The REF_VTG block uses the user-programmable DPLL for PCLK generation.

There are two options for PCLK generation:

1. Pre-defined frequencies:
   REFGEN_PREDEF_EN = 1

REFGEN_PREDEF_FREQ[1:0]	PREDEF_FREQ_ALT = 0	PREDEF_FREQ_ALT = 1
00	19.2MHz	13.5MHz
01	27MHz	24MHz
10	37.125MHz	27MHz
11	74.25MHz	27MHz

2. RCLK frequency by setting RCLKEN_Y = 1 for user-defined frequency generation (refer to the RCLKOUT Configuration section to for additional information).

The PCLK, HS, and VS signals generated by the REF_VTG block are synchronous; these can be output on MFP pins and routed to connected image sensor(s).

22.3.2 Key VTG Timing Parameters

Note: Reference the *VTG Configuration* section for comprehensive configuration details.

1. GEN_VS – Enable VS signal generation.
2. VS_INV – Invert the VS signal (default = positive sync polarity).
3. GEN_HS – Enable HS signal generation.
4. HS_INV – Invert the HS signal (default = positive sync polarity).
5. VS_DLY – Delay from the VS trigger to the generated VS signal expressed in PCLK cycles (default = 0).
6. VS_HIGH – VS high period in terms of PCLK cycles.
7. VS_LOW – VS low period in terms of PCLK cycles.
8. V2H – The delay from the VS trigger to the rising edge of the generated HS signal in terms of PCKL cycles.
9. HS_HIGH – HS high period in terms of PCLK cycles.
10. HS_LOW – HS low period in terms of PCLK cycles.
11. HS_CNT – HS counts/pulses per frame.
12. REFGEN_RST – Reset the REFGEN_PLL; used when changing the frequency.

ADI Confidential for Newvision under NDA
Analog Confidential Distribute Only to Maxim Integrated Company Confidential Customers Under NDA
Provided to GMSL2_Customers

22.4 Feature Availability by Device Family

Sync Pulse Outputs are available on all GMSL2 devices except the DSI Serializers, Advanced DSI Serializers, and Advanced CSI-2 Serializers (Table 76).

Table 76. Sync Pulse Outputs Support by Device Family

Device Family	Sync Pulse Outputs
HDMI Serializers	X
Advanced HDMI Serializers	X
DSI Serializers	
Advanced DSI Serializers	
CSI-2 Serializers	X
Advanced CSI-2 Serializers	
CSI-2 Camera Deserializers	X
CSI-2 Display Deserializers	X
OLDI Deserializers	X
eDP/DP Deserializers	X
CSI-2 Quad Deserializers	X

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Bidirectional Channels

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

23 Audio

23.1 Overview

GMSL2 devices support bidirectional transmission of audio (i.e., I2S and TDM) over the forward and reverse channels of the serial link. Audio signals received on the local side are tunneled over the serial link and reconstructed on the remote side. For system flexibility, GMSL2 devices can act as either the audio slave or the master at either end of the link. Reference *Feature Availability by Device Family* for audio capabilities of each GMSL2 device.

Traditional audio systems often comprise a mix of analog and digital transmission protocols. In these systems, analog interfaces at the input and output are bridged and processed digitally. This is illustrated below in *Figure 53*. I2S and TDM formats are typically used for inter-IC data communications and transfers on the same PC board, whereas off-board data connections may require other, long-distance protocols.

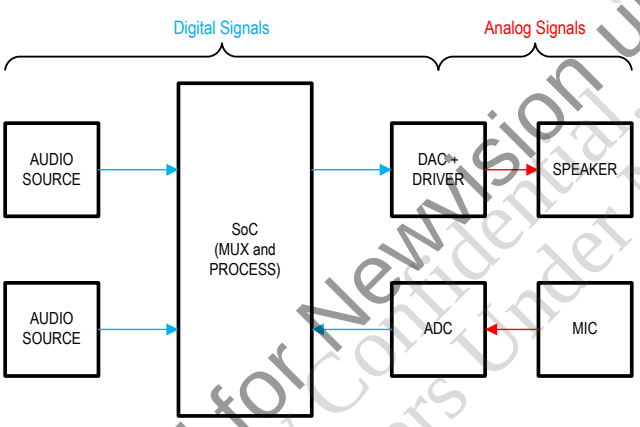


Figure 53. Typical Audio System Block Diagram

GMSL2 devices and the serial link serve as an integrated transmission interface for multiple data formats (e.g., video, audio, and control channel) within a system. Tunneling I2S/TDM audio data over the serial link eliminates the need for long-distance audio protocols (e.g., S/PDIF, optical, etc.) and simplifies the design of larger, interconnected systems (*Figure 54*).

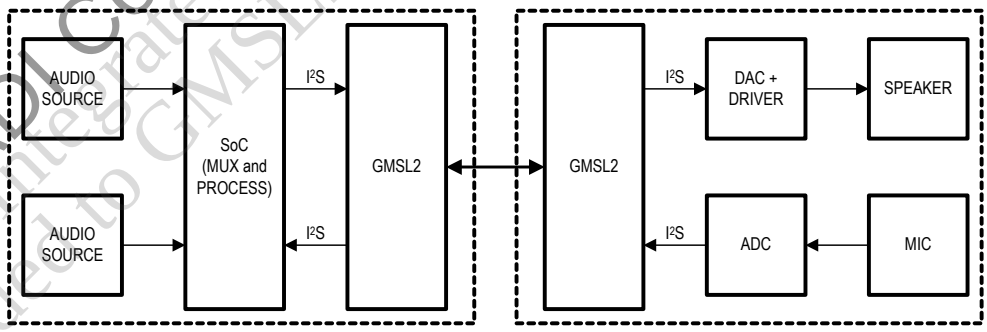


Figure 54. Typical GMSL2 Audio System Block Diagram

Analog Devices Confidential, Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA
ADI Confidential for New Version under NDA

23.2 Operation

23.2.1 Hardware Connection

An I2S interface consists of three pins: clock signal (SCK), data signal (SD), and word select (WS). Table 77 contains the I2S pin naming convention for forward and reverse audio.

Table 77. GMSL2 Device Audio Pinouts

GMSL2 Device	AUDIO Input Pins	AUDIO Output Pins
Serializer	SD/SCK/WS SD2/SCK2/WS2	SDOR/SCKOR/WSOR SDOR2/SCKOR2/WSOR2
Deserializer	SDIR/SCKIR/WSIR	SD/SCK/WS

Note: TDM audio uses the same pins as I2S audio.

23.2.2 Initialization

Configure and enable the audio block prior to initializing audio transmissions. When the configured audio block is enabled, the following operations are performed automatically upon device power-up.

Audio Master:
Upon detecting the presence an audio signal on the input pins:

- The number of SD bits per WS period are measured (when the reference clock stabilizes).
- The SCK clock frequency is measured.
- The SCK clock frequency and WS length are sent to the remote device using an info frame packet.
- Audio packets are sent over the link with the configured stream ID (there are four possible IDs). One audio packet is sent for each WS period.

Audio Slave:
Upon receiving audio packets over the GMSL link:

1. The audio receiver detects the presence of audio packets for the configured audio stream.
2. The info frame transmitted by the audio master (containing the SCK frequency and WS length) is received, and the audio output PLL initializes with the received SCK frequency.
3. The audio output SCK frequency is fine-tuned to match the detected input clock frequency using a FIFO control loop.
4. Audio is output from the enabled MFP pins.
5. If CRC/ARQ is enabled (default):
    a. **No CRC error**: The packet is sent to the audio receiver and an acknowledge is sent to the audio transmitter.
    b. **CRC error**: The packet is discarded and the retransmitted packet from the audio transmitter side is expected (see CRC Error Detection and ARQ Error Correction section for more info)

Analog Confidential – Limited Distribution Only – Under NDA
Maxim Integrated Confidential
Provided for GMSL2 Customers Under NDA
ADI Confidential for Newvision under NDA

23.2.3 Input Audio Properties

23.2.3.1 Data Format

GMSL2 devices support both I2S stereo and up to eight channels of audio in TDM mode. Data is tunneled from the local to the remote side of the serial link. Any data format that meets the length and frequency requirements can be used.

23.2.3.2 Length/Frequency

Sample rates (i.e., WS frequency) of 8kHz to 192kHz and sample depths of 8- to 32-bits are supported. For each WS cycle, there must be between 16 and 256 even-numbered bits. The SCK frequency range supported is 0.5MHz to 49.152MHz.

23.2.3.3 I2S vs TDM

Inter-IC Sound (I^2S) and Time-Division Multiplexing (TDM) are digital audio formats used to transmit audio data. Both formats use three signals (i.e., sync, clock, and data). I2S audio supports two-channel stereo (*Figure 55*), and TDM mode supports up to eight channels of audio (*Figure 56*). GMSL2 devices handle I2S and TDM formats similarly (except the WS polarity – reference the *Signal Polarity* section below). For system flexibility, GMSL2 devices (both serializers and deserializers) can act as either the audio slave or audio master at either end of the serial link.

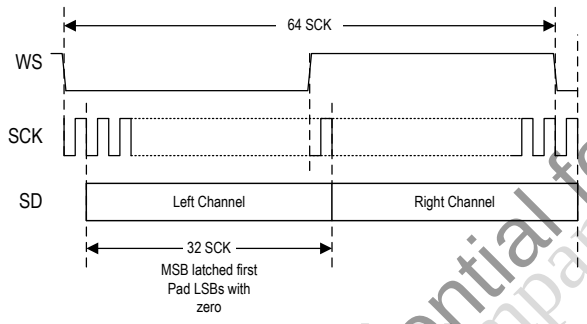


Figure 55. Two-Channel Stereo

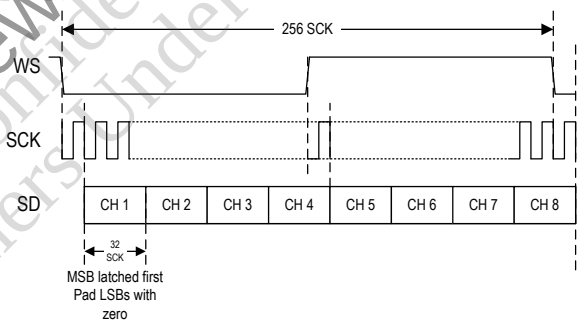


Figure 56. 8-Channel TDM

Note: The WS period is defined by the audio source and is not limited to the 50% duty cycle as shown in Figure 55 and Figure 56. However, the audio output duty cycle from the remote-side GMSL device is always 50%. Therefore, for WS inputs with a non-50% duty cycle, the audio output will not be a direct reproduction of the input.

In I2S format, two channels of audio are transmitted during each WS period; the audio data is transmitted in alternating channels starting with the MSB of the left channel. TDM format is similar to I2S format but includes up to eight channels of audio data.

Note: For I2S, the left and right channel designation are not limited to what is shown in *Figure 55*. For TDM mode, applications are not exclusively limited to 8-channel TDM as shown in *Figure 56*. The GMSL2 audio interface can be apply TDM mode in a range of possible channel configurations up to 8-channels.

ADI Confidential for NewVision Under NDA
Maxim Integrated Company Confidential, Customers Under NDA
Provided to GMSL2_

GMSL2 devices do not interpret the transmitted I2S data: the audio data is replicated on the opposing serializer or deserializer. The replicated data is interpreted by the processor in the receiver device on the opposing link. Ensure that the GMSL2 devices are configured properly to ensure that the transmitted audio data meets the requirements for the application.

23.2.3.4 Signal Polarity

GMSL2 systems allow for polarity configuration of the audio signal. By default, the SD and WS signals are sampled at the positive edge of the SCK (*Figure 57*). This behavior can be changed with the INV_SCK register in the AUDIO_TX block. The input polarity defaults to sampling audio frames at the falling edge of the WS signal. The INV_WS register in the AUDIO_TX block changes the tracking edge for the WS frame.

If a different polarity is used on the input, set the INV_SCK_TX or INV_WS_TX bits = 1 accordingly to use the reverse input polarity of the SCK or WS signal.

The WS and SCK input and output polarities can all be independently set. This allows for the device(s) to work with all polarity combinations on the input and output.

Note: TDM uses a rising edge of WS to indicate a start of frame. Bit INV_WS_TX must be set when using TDM audio as an input.

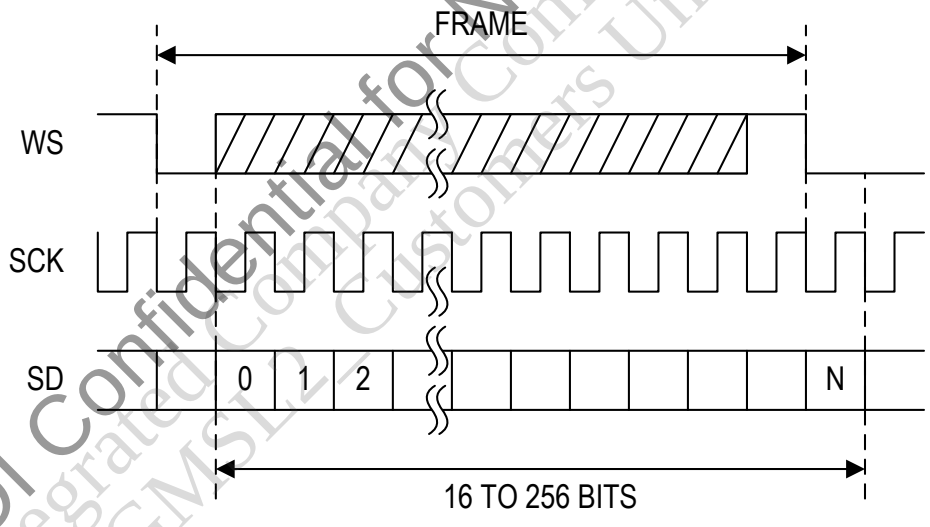


Figure 57. Default Input and Output Audio Polarity

ADI Confidential for Newvision under NDA
Analog Devices Inc. Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

23.2.4 Output Audio Properties

23.2.4.1 Signal Format

The output polarity defaults to sampling audio frames at the falling edge of the WS signal with data valid on the SCK rising edge. The INV_SCK_RX or INV_WS_RX registers in the AUDIO_RX block are used to independently change the SCK and WS output polarities. If the INV_WS register in the AUDIO_RX block is set, the frame data stays the same, but the frame boundary for audio samples is tracked at the rising edge of the WS signal.

The output WS will always have a 50% duty cycle regardless of input duty cycle.

Note: TDM uses a rising edge of WS to indicate a start of frame. Bit INV_WS_RX must be set when using TDM audio as an output.

23.2.4.2 MCLK

The MCLK (Master I2S Clock) is not transmitted through the GMSL2 link. If needed by the audio sink device, use either an externally generated source for the MCLK or configure the reference clock output (if available) to be used as the MCLK. See Clocks and Frequency Reference section for configuration details.

23.2.5 Audio Calculations

23.2.5.1 Audio Bandwidth Calculations

The GMSL2 link bandwidth consumption by the audio channel (either I2S or TDM) is given by the formulas below.

Audio Forward BW (Mbps) = $\left(\frac{\text{sample_rate}}{1000}\right) * 20 * \left(\text{roundup}\left(\text{channels} * \frac{\text{sample_depth}}{18} + 0.5\right) + 1 + \text{CRC}\right)$

Audio Reverse BW (Mbps) = $\left(\frac{sample_rate}{1000}\right) * 6 * 20 * \text{CRC}$

Where:

- sample_rate = audio sample rate (kbps).
- sample_depth = audio sample depth (bits per sample).
- channels = number of channels.
- roundup() is a function defined as rounding up the contained number to the next integer value.
- CRC = 1 if CRC is enabled; CRC = 0 if CRC is not enabled.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

23.2.5.2 Audio Clock Frequency Calculations

The audio clock frequencies for forward and reverse channel audio are calculated with the formulas given below.

Note: Convert all values to decimal for the equations.

Audio Clock Frequency (MHz) – Forward Channel Audio

Calculate the forward channel audio clock frequency with deserializer status register values.

$$Audio\ Clock\ (MHz) = \left(INFO_AUD_FB_DIV + \frac{INFO_AUD_FB_DIV_FRAC}{64}\right) * \left(\frac{75}{2 * INFO_AUD_ODIV * 2^{INFO_AUD_ODIV_EXP}}\right)$$

Audio Clock Frequency (MHz) – Reverse Channel Audio

Calculate the reverse channel audio clock frequency with serializer status register values.

$$Audio\ Clock\ (MHz) = \left(INFO_AUD_FB_DIV + \frac{INFO_AUD_FB_DIV_FRAC}{64}\right) * \left(\frac{75}{INFO_AUD_ODIV * 2^{INFO_AUD_ODIV_EXP}}\right)$$

Note: Reference the *GMSL2 Link Bandwidth Consumption from Side Channels* (*Audio*) section for more information.

23.2.6 HDCP and Audio

HDCP (High-bandwidth Digital Content Protection) protection of audio on GMSL links is a special use case. HDCP encryption operates with point-to-point data communication protocols. Since each audio stream is individually HDCP-encrypted, the protected audio stream cannot be broadcasted to multiple audio receivers. If broadcasting is required with HDCP enabled, two AUDIO_TX engines must be used to transmit the same audio signal as two separately encrypted data streams. All GMSL2 serializers have two AUDIO_TX modules (even if the device has only one set of I2S inputs) for this purpose. Reference the HDCP section for additional information.

ADI Confidential for Review under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL Customers Under NDA

23.3 Configuration

This section contains descriptions of the various operating modes and the register configurations required for programming. In each operating mode, the audio input is tunneled through the GMSL link and reconstructed on the audio output (except for any configured polarity inversion).

Modes of Operation

The following modes of operation and tests are supported by various GMSL2 devices. Refer to Feature Availability by Device Family or the part datasheet to confirm modes supported by device.

I2S/TDM Audio

1. Forward Channel Audio
2. Reverse Channel Audio
3. Forward and Reverse Channel Audio
4. Two Forward Channel Audio in Splitter Mode
5. Two Reverse Channel Audio in Splitter Mode
6. Four-Pin Bidirectional Audio I2S

HDMI Audio

7. HDMI Audio (HDMI serializers and Advanced HDMI serializers only)

PRBS Test Modes

8. Forward Channel Audio PRBS Test – Internal Oscillator (Table 86)
9. Forward Channel Audio PRBS Test – External Audio Clock (Table 87)
10. Reverse Channel Audio PRBS Test – Internal Oscillator (Table 88)
11. Reverse Channel Audio PRBS Test – External Audio Clock (Table 89)

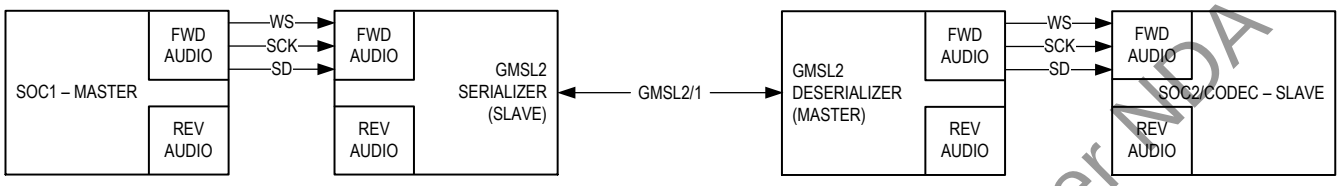
Note: Reference the *PRBS Testing* section for additional details.

ADI Confidential for review under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers under NDA

23.3.1 I2S/TDM Audio Configuration

23.3.1.1 Forward Channel Audio

Forward channel audio is serializer-to-deserializer audio (Figure 58). The serializer receives the audio on the I2S input pins and transmits the audio data over the serial link. The deserializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins. Each serializer has up to two forward audio channels (i.e., channel X and channel Y). Configuration is detailed in Table 78 and Table 79 below.



SERIALIZER SLAVE FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Input	Input, 1M Pulldown
WS	Input	Input, 1M Pulldown
SD	Input	Input, 1M Pulldown

DESERIALIZER MASTER FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Output	Push-Pull Driver
WS	Output	Push-Pull Driver
SD	Output	Push-Pull Driver

Figure 58. Forward Audio Channel I2S Block Diagram

Table 78. Configuration of Forward Channel Audio on Channel X

Description	Serializer Configuration	Deserializer Configuration
Enable audio transmission	Register: REG2 Bit: AUD_TX_EN_X = 1	-
Select Audio Stream ID – program serializer and deserializer to the same value	Register: Audio X: AUDIO_TX1 Bit: AUD_STR_TX[1:0]	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]
If TDM mode is used, invert WS	Register: AUDIO_TX0 Bit: INV_WS	-
Enable audio receiver	-	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1

Table 79. Configuration of Forward Channel Audio on Channel Y

Description	Serializer Configuration	Deserializer Configuration
Enable audio transmission	Register: REG2 Bit: AUD_TX_EN_Y = 1	-
Select Audio Stream ID – program serializer and deserializer to the same value	Register: Audio Y: AUDIO_TX1 Bit: AUD_STR_TX[1:0]	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]
If TDM mode is used, invert WS	Register: AUDIO_TX0 Bit: INV_WS	-
Enable audio receiver	-	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1

ADI Confidential or Newvision under NDA
Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA

23.3.1.2 Reverse Channel Audio

Reverse channel audio is deserializer-to-serializer audio (Figure 59). The deserializer receives the audio on the I2S input pins and transmits the audio data over the serial link. The serializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins. Forward and reverse audio channels are functionally the same, however, total available bandwidth is different. Reference Serial Link Rates by Part Number in GMSL2 Link Basics and/or device-specific datasheets for more information. Configuration is detailed in Table 80 below.

Note: The deserializer input I2S bus pin names are different than the serializer output I2S bus pin names. The deserializer uses the additional "I" to denote *Input*, whereas the serializer uses "O" to denote *Output* and "R" to indicate it is the *Reverse Channel*. Reference *Figure 59*.



SERIALIZER MASTER REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKOR	Output	Push-Pull Driver
WSOR	Output	Push-Pull Driver
SDOR	Output	Push-Pull Driver

DESERIALIZER SLAVE REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKIR	Input	Input, 1M Pulldown
WSIR	Input	Input, 1M Pulldown
SDIR	Input	Input, 1M Pulldown

Figure 59. Reverse Channel I2S Block Diagram

Table 80. Configuration of Reverse Channel Audio

Description	Serializer Configuration	Deserializer Configuration
Enable audio transmission	-	Register: REG2 Bit: AUD_TX_EN = 1
Select Audio Stream ID – program serializer and deserializer to the same value	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]	Register: AUDIO_TX1 Bit: AUD_STR_TX[1:0]
If TDM mode is used, invert WS	-	Register: AUDIO_TX0 Bit: INV_WS
Enable audio receiver	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1	-

ADI Confidential for New Version under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

23.3.1.3 Forward and Reverse Channel Audio

Forward and reverse channel audio is the simultaneous transmission of both serializer-to-deserializer and deserializer-to-serializer audio (Figure 60). On the forward channel, the serializer receives the audio on the I2S input pins and transmits the audio data over the serial link. The deserializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins. On the reverse channel, the deserializer receives the audio on the I2S input pins and transmits the audio data over the serial link. The serializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins. The configuration procedure to enable both forward and reverse channel audio at the same time is presented in Table 81 below.



SERIALIZER SLAVE FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Input	Input, 1M Pulldown
WS	Input	Input, 1M Pulldown
SD	Input	Input, 1M Pulldown

DESERIALIZER MASTER FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Output	Push-Pull Driver
WS	Output	Push-Pull Driver
SD	Output	Push-Pull Driver

SERIALIZER MASTER REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKOR	Output	Push-Pull Driver
WSOR	Output	Push-Pull Driver
SDOR	Output	Push-Pull Driver

DESERIALIZER SLAVE REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKIR	Input	Input, 1M Pulldown
WSIR	Input	Input, 1M Pulldown
SDIR	Input	Input, 1M Pulldown

Figure 60. Forward and Reverse Channel I2S Block Diagram

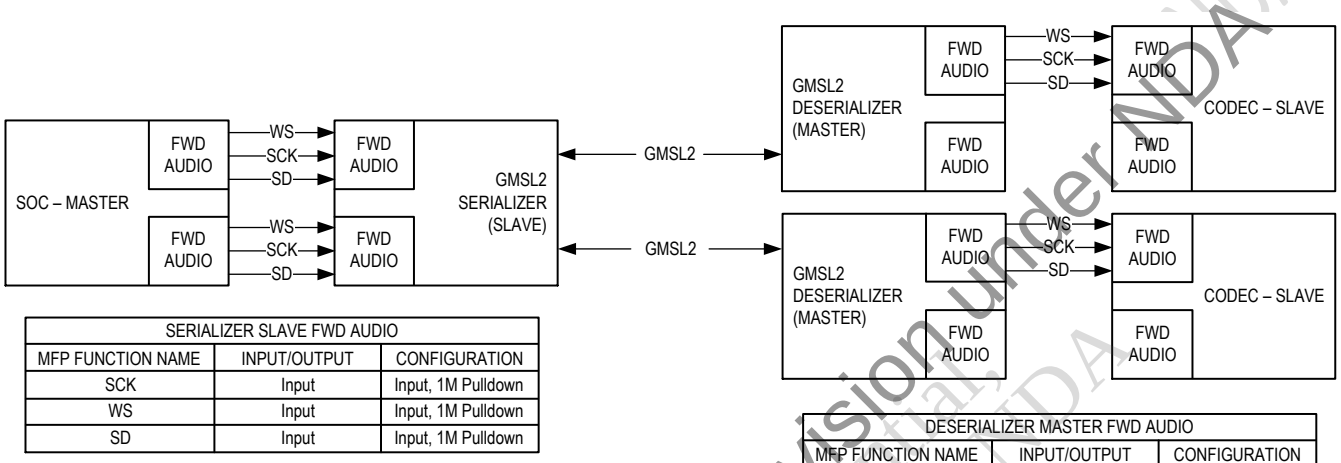
Table 81. Forward and Reverse Channel Audio

Description	Serializer Configuration	Deserializer Configuration
Enable forward channel audio transmitter	Register: REG2 Bit: AUD_TX_EN_X = 1	-
Enable forward channel audio receiver	-	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1
Enable reverse channel audio transmitter	-	Register: REG2 Bit: AUD_TX_EN = 1
Enable reverse channel audio receiver	Register: Audio X: AUDIO_RX1 Bit: AUD_EN_RX = 1	
Select Audio Stream ID for Forward Channel Audio – program serializer and deserializer to the same value	Register: Audio X: AUDIO_TX1 Bit: AUD_STR_TX[1:0]	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]
Select Audio Stream ID for Reverse Channel Audio – program serializer and deserializer to the same value	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]	Register: AUDIO_TX1 Bit: AUD_STR_TX[1:0]
Only for DSI Serializers: Select which PHY to receive reverse channel audio from	Register: REG3 Bit: AUD_RX_A_SRC = 0 for PHY A Bit: AUD_RX_A_SRC = 1 for PHY B	-
Only for DSI Serializers: Select which audio input pins to use for forward channel audio input	Register: REG4 Bit: AUD_TX_SRC_X = 0 for SCK,SD,WS Bit: AUD_TX_SRC_X = 1 for SCK2,SD2,WS2	-

ADI Confidential for Newvision under NDA
Maxim Confidential
Provided to GMSL2_Customers Under NDA

23.3.1.4 Two Forward Channel Audio in Splitter Mode

Two forward channel audio in splitter mode is serializer-to-deserializer audio (Figure 61). A single serializer in splitter mode (via two GMSL2 PHYs) connects to two deserializers. The serializer receives the audio on the I2S input pins and transmits the audio data over the serial link. Both deserializers receive the audio data from the serial link, independently reconstruct the audio, and output the data to their respective I2S pins. The available bandwidth to each deserializer is equal to the forward channel link rate that has be configured. Configuration is detailed in Table 82 below.



SERIALIZER SLAVE FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Input	Input, 1M Pulldown
WS	Input	Input, 1M Pulldown
SD	Input	Input, 1M Pulldown

DESERIALIZER MASTER FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Output	Push-Pull Driver
WS	Output	Push-Pull Driver
SD	Output	Push-Pull Driver

Figure 61. Two Forward Channel I2S in Splitter Mode Block Diagram

Table 82. Two Forward Channel Audio in Splitter Mode

Description	Serializer Configuration	Deserializer 1 Config	Deserializer 2 Config
Send Audio X packets to GMSL PHY A	Register: AUDIO_X: TR3 Bit: TX_SPLT_MASK_A = 1	-	-
Do not send Audio X packets to GMSL PHY B	Register: AUDIO_X: TR3 Bit: TX_SPLT_MASK_B = 0	-	-
Do not send Audio Y packets to GMSL PHY B	Register: AUDIO_Y: TR3 Bit: TX_SPLT_MASK_A = 0	-	-
Send Audio Y packets to GMSL PHY B	Register: AUDIO_Y: TR3 Bit: TX_SPLT_MASK_B = 1	-	-
Audio channel X: Set ARQ settings to manual mode	Register: AUDIO_X: ARQ0 Bit: ARQ_AUTO_CFG = 0	-	-
Audio channel X: Use TX_SRC_ID bitfield instead of SRC_ID of received data packet	Register: AUDIO_X: ARQ0 Bit: ACK_SRC_ID = 1	-	-
Audio channel X: Ack packet is accepted if SRC_ID of the received Ack packet matches TX_SRC_ID register	Register: AUDIO_X: ARQ0 Bit: MATCH_SRC_ID = 1	-	-
Audio channel X: Wait for one ACK instead of two	Register: AUDIO_Y: ARQ0 Bit: ACK_CNT = 0	-	-

ADI Confidential for Review under NDA
Maxim Integrated Confidential, Provided to Customers Under NDA

Audio channel Y: Set ARQ settings to manual mode	Register: AUDIO_Y: ARQ0 Bit: ARQ_AUTO_CFG = 0	-	-
Audio channel Y: Use TX_SRC_ID bitfield instead of SRC_ID of received data packet	Register: AUDIO_Y: ARQ0 Bit: ACK_SRC_ID = 1	-	-
Audio channel Y: Ack packet is accepted if SRC_ID of the received Ack packet matches TX_SRC_ID register	Register: AUDIO_Y: ARQ0 Bit: MATCH_SRC_ID = 1	-	-
Audio channel Y: Wait for one ACK instead of two	Register: AUDIO_Y: ARQ0 Bit: ACK_CNT = 0	-	-
Audio channel Y: Enable ARQ (not enabled by default ton Pipe Y)	Register: AUDIO_Y: ARQ0 Bit: ARQ_EN = 1	-	-
Deserializers: Set ARQ settings to manual mode	-	Register: AUDIO: ARQ0 Bit: ARQ_AUTO_CFG = 0	Register: AUDIO: ARQ0 Bit: ARQ_AUTO_CFG = 0
Deserializers: Use SRC_ID of the received data packet	-	Register: AUDIO: ARQ0 Bit: ACK_SRC_ID = 0	Register: AUDIO: ARQ0 Bit: ACK_SRC_ID = 0
Deserializers: All received Ack packets are accepted	-	Register: AUDIO: ARQ0 Bit: MATCH_SRC_ID = 0	Register: AUDIO: ARQ0 Bit: MATCH_SRC_ID = 0
Deserializers: Wait for one ACK instead of two	-	Register: AUDIO: ARQ0 Bit: ACK_CNT = 0	Register: AUDIO: ARQ0 Bit: ACK_CNT = 0
Audio Channel X: Transmit audio packets on stream id 0	Register: Audio X: AUDIO_TX1 Bit: AUD_STR_TX[1:0] = 00	-	-
Audio Channel Y: Transmit audio packets on stream id 1	Register: Audio Y: AUDIO_TX1 Bit: AUD_STR_TX[1:0] = 01	-	-
Set deserializers to Receive on audio Stream ID 0, and 1 respectively	-	Register: AUDIO_RX7 Bit: AUD_STRM[1:0] = 00	Register: AUDIO_RX7 Bit: AUD_STRM[1:0] = 01
Transmit Audio with STREAM ID 0 from Channel X	Register: Audio X: AUDIO_TX1 Bit: AUD_STR_TX[1:0] = 00	-	-
Transmit Audio with STREAM ID 1 from Channel Y	Register: Audio Y: AUDIO_TX1 Bit: AUD_STR_TX[1:0] = 01	-	-
Select audio input pins for audio channel X. Set AUD_TX_SRC_X depending on which pins are desired	Register: REG4 Bit: AUD_TX_SRC_X	-	-
Select audio input pins for audio channel Y. Set AUD_TX_SRC_Y depending on which pins are desired	Register: REG4 Bit: AUD_TX_SRC_Y	-	-
Disable reverse channel audio on deserializers	-	Register: REG2 Bit: AUD_TX_EN = 0	Register: REG2 Bit: AUD_TX_EN = 0
Disable reverse channel audio channel X	Register: Audio X: AUDIO_RX1 Bit: AUD_EN_RX = 0	-	-

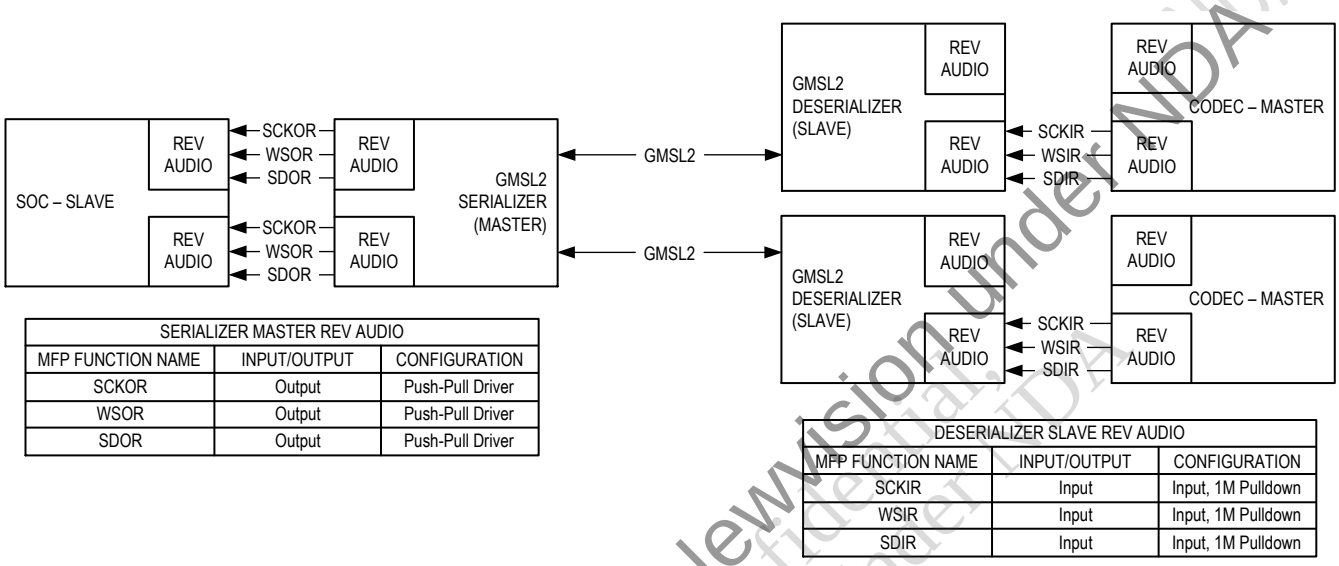
Maxim Confidential for Newvision under NDA
Maxim Proprietary Confidential
Provided to GMSL2 Customers Under NDA

Disable reverse channel audio channel Y	Register: Audio Y: AUDIO_RX1 Bit: AUD_EN_RX = 0	-	--
Forward channel audio enable in deserializers	-	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1
If TDM mode is used on channel X, invert WS	Register: AUD_TX__AX: AUDIO_TX0 Bit: INV_WS	-	-
If TDM mode is used on channel Y, invert WS	Register: AUD_TX__AY: AUDIO_TX0 Bit: INV_WS	-	-
Audio transmitter X enable	Register: REG2 Bit: AUD_TX_EN_X = 1	-	-
Audio transmitter Y enable	Register: REG2 Bit: AUD_TX_EN_Y = 1	-	-
Reset link	Register: CTRL0 Bit: RESET_ONESHOT = 1	-	-

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute Only Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

23.3.1.5 Two Reverse Channel Audio in Splitter Mode

Two reverse channel audio in splitter mode is deserializer-to-serializer audio (Figure 62). Two deserializers connect to a single serializer in splitter mode (via two GMSL2 PHYs). The deserializers receive the audio on the respective I2S input pins and independently transmit the audio data over the serial link. The serializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins. The available bandwidth from each deserializer is equal to the reverse channel link rate that has been configured. Configuration is detailed in Table 83 below.



SERIALIZER MASTER REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKOR	Output	Push-Pull Driver
WSOR	Output	Push-Pull Driver
SDOR	Output	Push-Pull Driver

DESERIALIZER SLAVE REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKIR	Input	Input, 1M Pulldown
WSIR	Input	Input, 1M Pulldown
SDIR	Input	Input, 1M Pulldown

Figure 62. Two Reverse Channel I2S in Splitter Mode Block Diagram

Table 83. Reverse Channel Audio

Description	Serializer Configuration	Deserializer 1 Config	Deserializer 2 Config
Packets NOT transmitted over GMSLA in splitter	Register: AUDIO_Y: TR3 Bit: TX_SPLT_MASK_A = 0	-	-
Packets ARE transmitted over GMSLB in splitter	Register: AUDIO_Y: TR3 Bit: TX_SPLT_MASK_B = 1	-	-
Packets are transmitted over GMSLA in splitter	Register: AUDIO_X: TR3 Bit: TX_SPLT_MASK_A = 1	-	-
Packets NOT transmitted over GMSLB in splitter	Register: AUDIO_X: TR3 Bit: TX_SPLT_MASK_B = 0	-	-
Set Audio Tx stream id 0 and 1 respectively	-	Register: AUDIO_TX1 Bit: AUD_STR_TX[1:0] = 00	Register: AUDIO_TX1 Bit: AUD_STR_TX[1:0] = 01
Audio transmitter X disable	Register: REG2 Bit: AUD_TX_EN_X = 0	-	-
Audio transmitter Y disable	Register: REG2 Bit: AUD_TX_EN_Y = 0	-	-
Audio Receiver X enable	Register: Audio X: AUDIO_RX1 Bit: AUD_EN_RX = 1	-	-
Audio Receiver Y enable	Register: Audio Y: AUDIO_RX1 Bit: AUD_EN_RX = 1	-	-
If TDM mode is used, invert WS	-	Register: AUDIO_TX0 Bit: INV_WS	Register: AUDIO_TX0 Bit: INV_WS

Maxim Confidential for New Vision under NDA
Provided to GMSL Customers Under NDA
Maxim Integrated Company Confidential

Select SRC ID to receive on X	Register: AUDIO_X: TR4 Bit: RX_SRDC_SEL[7:0] = All zeroes except the bit with index equal to DESB's TX_SRC_ID_AUDIO	-	-
Select SRC ID to receive on X	Register: AUDIO_Y : TR4 Bit: RX_SRDC_SEL[7:0] = All zeroes except the bit with index equal to DESB's TX_SRC_ID_AUDIO	-	-
ARQ settings for X are based on selections in ACK_SRC_ID and MATCH_SRC_ID bitfields	Register: AUDIO_X: ARQ0 Bit: ARQ_AUTO_CFG = 0	-	-
Wait for 1 ACK instead of 2 on X	Register: AUDIO_X: ARQ0 Bit: ACK_CNT_AUDIO_X = 0	-	-
All received Ack packets are accepted	Register: AUDIO_X: ARQ0 Bit: MATCH_SRC_ID = 0	-	-
Use TX_SRC_ID bitfield instead of SRC_ID for X	Register: AUDIO_X: ARQ0 Bit: ACK_SRC_ID = 1	-	-
ARQ settings for Y are based on selections in ACK_SRC_ID and MATCH_SRC_ID bitfields	Register: AUDIO_Y: ARQ0 Bit: ARQ_AUTO_CFG = 0	-	-
Wait for 1 ACK instead of 2 on Y	Register: AUDIO_Y: ARQ0 Bit: ACK_CNT_AUDIO_X = 0	-	-
All received Ack packets are accepted	Register: AUDIO_Y: ARQ0 Bit: MATCH_SRC_ID = 0	-	-
Use TX_SRC_ID bitfield instead of SRC_ID for Y	Register: AUDIO_Y: ARQ0 Bit: ACK_SRC_ID = 1	-	-
Select Audio channel to receive audio from PHYA	Register: REG3 Bit: AUD_RX_A_SRC = 0 (Toggle to swap pins on SER output)	-	-
Select Audio channel to receive audio from PHYB	Register: REG3 Bit: AUD_RX_B_SRC = 0 (Toggle to swap pins on SER output)	-	-
Receive audio stream ID 0 on A	Register: AUDIO_RX7 Bit: AUD_STRM[1:0] = 00	-	-
Receive audio stream ID 1 on B	Register: AUDIO_RX7 Bit: AUD_STRM[1:0] = 01	-	-
Audio Receiver disable	-	Register: AUDIO_RX1 Bit: AUD_EN_RX = 0	Register: AUDIO_RX1 Bit: AUD_EN_RX = 0
Audio transmitter enable	-	Register: REG2 Bit: AUD_TX_EN = 1	Register: REG2 Bit: AUD_TX_EN = 1
Reset link	Register: CTRL0 Bit: RESET_ONESHOT = 1	-	-

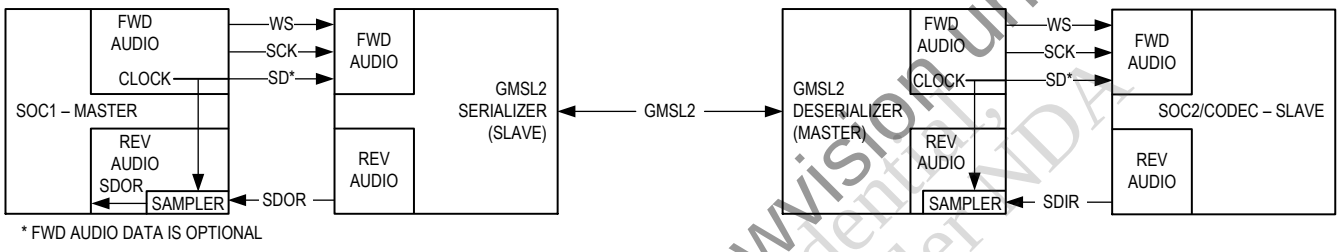
ADI Confidential for Newvision under NDA
Maxim Integrated Provided to GMSL2_Customers Under NDA

23.3.1.6 Four-Pin Bidirectional Audio I2S

When the serial link audio configuration is the same for both audio directions, it is possible to use only the SD pin (and eliminate the WS and SCK pins) for one of the directions to reduce the system pin count from six to four. Here, the SD pin is clocked off the SCK of the opposite channel.

On the forward channel, the serializer receives the audio on the I2S input pins and transmits the audio data over the serial link. The deserializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins. On the reverse channel, the deserializer receives the audio on I2S input pins and transmits the audio data over the serial link. The serializer receives the audio data from the serial link, reconstructs the audio, and outputs the data to the I2S pins.

The diagrams given below illustrate the system when forward channel (Figure 63) or the reverse channel (Figure 64) are designated as the master for the bidirectional 4-pin audio system. Configuration for the forward channel is detailed in Table 84 and for the reverse channel in Table 85 below.



* FWD AUDIO DATA IS OPTIONAL

SERIALIZER SLAVE FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Input	Input, 1M Pulldown
WS	Input	Input, 1M Pulldown
SD	Input	Input, 1M Pulldown

DESERIALIZER MASTER FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Output	Push-Pull Driver
WS	Output	Push-Pull Driver
SD	Output	Push-Pull Driver

SERIALIZER SLAVE REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SDOR	Output	Push-Pull Driver
SCKOR	Not Used	N/A
WSOR	Not Used	N/A

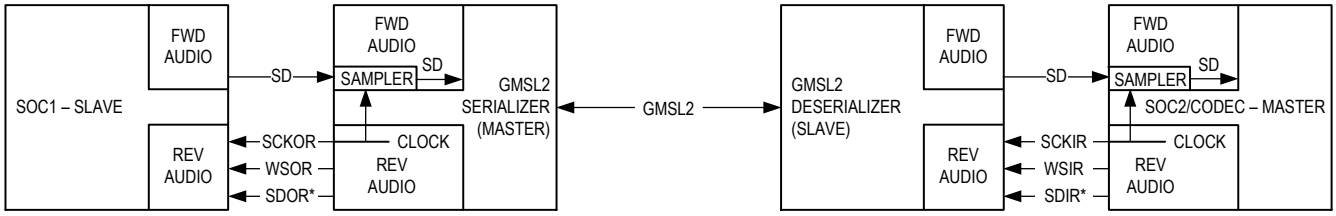
DESERIALIZER MASTER REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKIR	Input	Input, 1M Pulldown
WSIR	Not Used	N/A
SDIR	Not Used	N/A

Figure 63. Forward Audio Master Set: AUD_SINK_SRC (Deserializer) and AUD_RX_SINK_SRC (Serializer)

Table 84. Configuration for Four-Pin Bidirectional Audio – Forward Audio Master

Description	Serializer Configuration	Deserializer Configuration
Enable forward channel audio transmission	Register: REG2 Bit: AUD_TX_EN = 1	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1
If TDM mode is used, invert WS	Register: AUDIO_TX0 Bit: INV_WS	Register: AUDIO_TX0 Bit: INV_WS
Select Audio Stream ID – program serializer and deserializer to the same value	Register: Audio X: AUDIO_TX1 Bit: AUD_STR_TX[1:0]	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]
Enable reverse channel audio transmission	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1	Register: REG2 Bit: AUD_TX_EN = 1
Enable bidirectional sink sourced mode for the reverse channel audio	Register: AUDIO_RX1 Bit: AUD_RX_SINK_SRC = 1	Register: AUDIO_TX0 Bit: AUD_SINK_SRC = 1

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA



* REV AUDIO IS OPTIONAL

SERIALIZER MASTER FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Input	Input, 1M Pulldown
WS	Not Used	N/A
SD	Not Used	N/A

DESERIALIZER SLAVE FWD AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCK	Output	Push-Pull Driver
WS	Not Used	N/A
SD	Not Used	N/A

SERIALIZER MASTER REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SDOR	Output	Push-Pull Driver
SCKOR	Output	Push-Pull Driver
WSOR	Output	Push-Pull Driver

DESERIALIZER SLAVE REV AUDIO		
MFP FUNCTION NAME	INPUT/OUTPUT	CONFIGURATION
SCKIR	Input	Input, 1M Pulldown
WSIR	Input	Input, 1M Pulldown
SDIR	Input	Input, 1M Pulldown

Figure 64. Reverse Audio Master Set: AUD_SINK_SRC (Serializer) and AUD_RX_SINK_SRC (Deserializer)

Table 85. Configuration for Four-Pin Bidirectional Audio – Reverse Audio Master

Description	Serializer Configuration	Deserializer Configuration
Enable forward channel audio transmission	Register: REG2 Bit: AUD_TX_EN = 1	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1
If TDM mode is used, invert WS	Register: AUDIO_TX0 Bit: INV_WS	Register: AUDIO_TX0 Bit: INV_WS
Select Audio Stream ID – program serializer and deserializer to the same value	Register: Audio X: AUDIO_TX1 Bit: AUD_STR_TX[1:0]	Register: AUDIO_RX7 Bit: AUD_STRM[1:0]
Enable reverse channel audio transmission	Register: AUDIO_RX1 Bit: AUD_EN_RX = 1	Register: REG2 Bit: AUD_TX_EN = 1
Enable bidirectional sink sourced mode for the forward channel audio	Register: AUDIO_TX0 Bit: AUD_SINK_SRC = 1	Register: AUDIO_RX1 Bit: AUD_RX_SINK_SRC = 1

ADI Confidential for NewVision under NDA
Analog Devices Confidential
Maxim Integrated Confidential,
Provided to GMSL2_Customers Under NDA

23.3.2 HDMI Audio Configuration

GMSL2 HDMI serializers and advanced HDMI serializers use HDMI audio for the serial link audio channel if audio is present on the HDMI input. For consumer port applications, HDMI audio capability must be added to the EDID table, and the source device settings must have 'HDMI display' selected as the audio output device. For embedded applications, the SoC must enable its HDMI audio output. See the HDMI Serializers section for additional information.

23.3.2.1 HDMI Audio

In order to set the HDMI as the audio source, toggle the reg_i2s_mode[0] bit in register 0x2129 to '1' along with the basic I2S forward audio setup.

23.3.2.1.1 HDMI Source – Forward Audio Example

Serializer Setup:

1. 0x80,0x2129,0x1C # reg_i2s_mode[0] = 1
2. 0x80,0x0002,0x47 # AUD_TX_EN_X[0] = 1
3. 0x80,0x0120,0x00 # AUD I2S_CFG[0-1] = 0

Deserializer Setup:

4. 0x54,0x0140,0x21 # AUD_EN_RX[0] = 1
5. 0x54,0x01CE,0x5E # OLDI Display Setup = 1
6. 0x80,0x0001,0x88 # SER HDMI_AUTOS[0] = 1

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

23.3.3 Audio PRBS Tests

GMSL2 devices can perform various audio pseudo-random binary sequence (PRBS) tests that can be used to test and debug the audio system. These tests can also be performed to determine the bit error rate (BER) of audio packets. See the PRBS Testing section for additional information (including other PRBS tests available).

Table 86. Forward Channel Audio PRBS Test – Internal Oscillator

Description	Serializer Configuration	Deserializer Configuration
Enable audio receiver	-	AUD_EN_RX = 1'b1
Enable audio transmission (Channel X)	AUD_TX_EN_X = 1'b1	-
Use the internal oscillator clock (150MHz) for PRBS test	PRBS_SEL = 1'b1	-
Enable audio PRBS test	PRBSEN_AUD = 1'b1	-
Enable audio PRBS checker	-	APRBS_CHK_EN = 1'b1

Table 87. Forward Channel Audio PRBS Test – External Audio Clock

Description	Serializer Configuration	Deserializer Configuration
Enable audio receiver	-	AUD_EN_RX = 1'b1
Enable audio transmission (Channel X)	AUD_TX_EN_X = 1'b1	-
Use SCK for PRBS test	PRBS_SEL = 1'b0	-
Enable audio PRBS test	PRBSEN_AUD = 1'b1	-
Enable audio PRBS checker	-	APRBS_CHK_EN = 1'b1

Table 88. Reverse Channel Audio PRBS Test – Internal Oscillator

Description	Serializer Configuration	Deserializer Configuration
Enable audio receiver (Channel X)	AUD_EN_RX_X = 1'b1	-
Enable audio transmission	-	AUD_TX_EN = 1'b1
Use the internal oscillator clock (150MHz) for PRBS test	-	PRBS_SEL = 1'b1
Enable audio PRBS test	-	PRBSEN_AUD = 1'b1
Enable audio PRBS checker	APRBS_CHK_EN = 1'b1	-

Table 89. Reverse Channel Audio PRBS Test – External Audio Clock

Description	Serializer Configuration	Deserializer Configuration
Enable audio receiver (Channel X)	AUD_EN_RX_X = 1'b1	-
Enable audio transmission	-	AUD_TX_EN = 1'b1
Use SCK for PRBS test	-	PRBS_SEL = 1'b0 to use SCK
Enable audio PRBS test	-	PRBSEN_AUD = 1'b1
Enable audio PRBS checker	APRBS_CHK_EN = 1'b1	-

Maxim Confidential
Provided to GMSL2_Customers Under NDA
Maxim Confidential
Provided to GMSL2_Customers Under NDA
Maxim Confidential for Newvision under NDA

23.4 Audio Status and Debug

This section explains how to check for audio errors and provides debugging procedures for unexpected behavior.

23.4.1 Status Registers

The following status registers are used to ensure the proper implementation of audio in a system.

ACLKDET (register AUDIO_TX5) – Indicates if an audio input clock is detected. If this is 0, check that the audio system is enabled with the AUD_TX_EN registers, the audio input pins are set up and multiplexed to the AUDIO TX block, and valid audio input is applied. Ensure that there is adequate bandwidth for the audio transformer by confirming that the AUD_OVERFLOW register is 0.

AUD_LOCK (register AUDIO_RX9) – Real-time flag indicating if the audio receiver output is operational. Check that the AUD_EN_RX is enabled and that the proper audio stream is selected from AUD_STRM to ensure that the audio receiver sub-system is enabled.

AUD_PKT_DET (register AUDIO_RX9) – Real-time flag indicating if the audio receiver is receiving the desired audio packets properly. Check that the AUD_EN_RX is enabled and that the proper audio stream is selected from AUD_STRM to verify that the audio receiver sub-system is enabled.

Registers ACLKDET[0], AUD_LOCK[0], and AUD_PKT_DET[0] aid during the I2S audio debugging process. If there is no audio on the receiver side, check if the register ACLKDET[0] is high. If ACLKDET[0] is low, then there is no audio clock detected at the receiver side. Ensure that the I2S receiver side has the AUD_EN_RX[0] bit set and the transmitter side has the AUD_TX_EN[0] bit set.

Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
ADI Confidential for Newvision under NDA

23.4.2 Debug Registers

- AUD_DRIFT_ERR – Audio clock drift error. The audio transmitter automatically restarts the audio transmission and reports an error if the audio clock drifts excessively.
- ACLKDET – Audio clock stop. If audio clock stops even for a short duration, this event is detected and flagged by the transmitter side clock detector and audio subsystem restarts.
- AUD_BLK_LEN_ERR – Audio receiver block length error. The receiver side audio receiver automatically restarts if the audio clock frequency output fine tuning FIFO overflows or underflows due to errors. This can happen due to errors on the link especially when ARQ is turned off.

The following device audio error registers can be used for error diagnostics:

- APRBS_ERR[0-7] Number of audio PRBS errors detected; clears on read.
- AUD_FIFO_WARN[0] Audio FIFO is more than half full if high.
- AUD_OVERFLOW[0] Audio FIFO buffer overflow detected if high.
- AUD_DRIFT_ERR[0] Audio Clock frequency drift detected if high.

23.4.3 Debug Procedure

Perform the following procedure to debug the audio system:

- Check that the audio input connected to the correct input pins, exhibiting the expected IOVDD voltage level, and has been started.
- Ensure that the audio input format is stable and compliant with I2S specs and the datasheet EC table.
- Verify that both the audio receiver and transmitter are enabled.
- Confirm that both ends of the link are using the same audio stream ID.
- Check that there are no errors or retransmission events on the link via the RT_CNT and DEC_ERR registers. This could indicate poor link quality causing audio packet corruption.
- Ensure that the audio stream is routed to the correct GMSL PHY with the TX_SPLT_MASK_A/B_AUD registers.
- Verify that the debug/status/error registers (above) are at their expected values.
- Toggle ARQ to check if a configuration problem in ARQ sub-system is preventing audio from functioning properly.
- Verify that there is sufficient link bandwidth available for the audio stream.

ADI Confidential Under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

23.5 Feature Availability by Device Family

All GMSL2 devices have audio capabilities except some CSI-2 parts. The physical connections and capabilities may vary by device family. See *Table 90* for details.

Table 90. Audio Support by Device Family

Device Family	I2S Input Pin Sets	I2S Output Pin Sets	Audio Tx Blocks	Audio Rx Blocks	HDMI Audio
HDMI Serializers	1	1	2	1	Yes
Advanced HDMI Serializers	1	1	2	1	Yes
DSI Serializers	1	1	2	1	No
Advanced DSI Serializers	2	2	2	2	No
CSI-2 Serializers	0	0	0	0	No
Advanced CSI-2 Serializers	0	0	0	0	No
CSI-2 Camera Deserializers	0	0	0	0	No
CSI-2 Display Deserializers	1	1	1	1	No
OLDI Deserializers	1	1	1	1	No
eDP/DP Deserializers	1	1	1	1	No
CSI-2 Quad Deserializers	0	0	0	0	No

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

24 I2C/UART

24.1 Overview

I2C and UART are serial communication protocols supported by all GMSL2 devices. The main control channel (CC) and pass-through (tunneling) channels utilize these protocols to transmit control information from the host controller to the GMSL2 devices and connected peripherals from either end of the serial link.

The main CC provides access to both the internal registers of GMSL2 devices and connected peripheral devices; the pass-through I2C/UART channels provide a direct connection to remote I2C/UART peripherals but do not provide access to internal GMSL2 device registers.

24.2 Main Control Channel – I2C/UART

24.2.1 Overview

The main I2C/UART control channel (CC) provides a master microcontroller (µC) with access to GMSL2 device registers and connected peripheral devices from either end of the serial link. Typical applications use one master µC (Figure 65). I2C multi-master applications are also supported, provided that a software arbitration method is used to prevent collisions (*Figure 66*). Note that the serial link allows only one master µC to communicate at any given time. The main CC also enables the master µC to configure peripherals connected at the other end of the serial link (i.e., remote side). This application requires that both GMSL2 devices are configured for the same CC protocol (i.e., both I2C or both UART); GMSL2 devices do not support I2C-to-UART or UART-to-I2C conversion.

Note: In CC applications, the local GMSL2 device is connected to the µC; the remote GMSL2 device is connected to the remote peripheral I2C/UART device (see *Figure 65*).

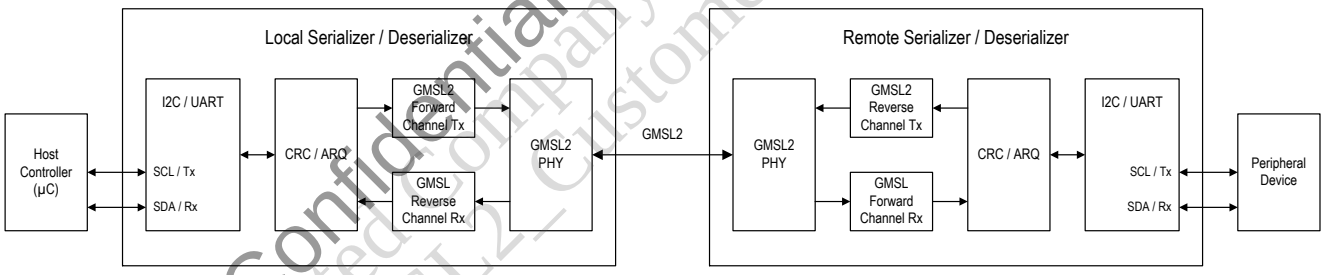


Figure 65. Typical GMSL2 Serial Link System with Main Control Channel

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

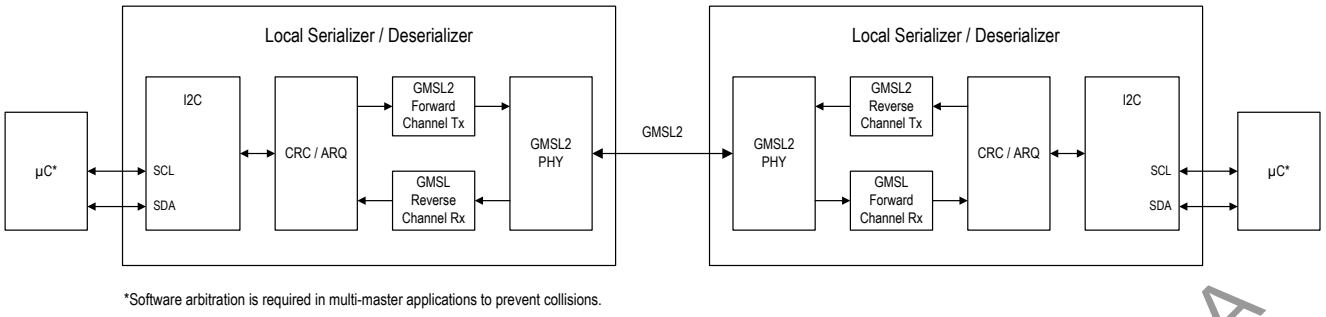


Figure 66. GMSL2 System with Multi-Master Main Control Channel (I2C)

Both CC protocols use the GMSL2 device's SDA_RX and SCL_TX pins. Selection of I2C (SDA and SCL) or UART (Rx and Tx) is determined at power-up with the I2CSEL pin state or the Configuration Pin (CFG). Refer to device-specific datasheet for the protocol selection settings; note that some parts may feature pins with shared functionality. The I2C/UART outputs are open-drain and require external pullup resistors to ensure proper operation. The value of the external pullup resistors is application-specific and is determined by the characteristics of the connected µC or other peripherals and the value of V_{DDIO}; reference applicable documentation for additional information.

24.2.2 Operation

All GMSL2 device registers are accessible with an external µC via the main control channel. GMSL2 devices use 16-bit register addressing. Protocol details are provided below (see *I2C Data Transfer Format* and *UART Frame Format*).

Note: GMSL1 devices use 8-bit register addressing; GMSL2 devices use 16-bit register addressing.

The main CC protocol is selected by setting the I2CSEL pin or CFG pins to the required value with a resistor network. The pin level is latched at power-up and selects either I2C or UART operation. Reference device-specific datasheet for configuration details.

The GMSL2 device directly connected to the µC that programs the serial link system and accesses remote peripherals through the main CC is called the local device. The GMSL2 device on the opposite side of the serial link of the local device is called the remote device. Reference Figure 65.

Remote-side devices are connected to the main CC of local GMSL2 device when the link is locked. To configure remote devices over the serial link, both the serializer and deserializer must be configured to the same protocol (i.e., I2CSEL or CFG settings must match for both devices).

Each device on the I2C/UART CC must have a unique device address that is used as the I2C/UART slave address. The voltage level of the ADD or CFG pins (depending on availability) at power-up sets the initial GMSL2 device address and the default value of the TX_SRC_ID registers for bidirectional channels. After power-up, the GMSL2 device address can be changed by writing to the DEV_ADDR register. If a device address is changed, all subsequent transactions with the device must use the new address.

ADI Confidential for New Vision under NDA
Maxim Official Distributor
Provided to GMSL2 Customers Under NDA
Analog Devices Confidential, Integrated Company Confidential,

In general, only one µC (I2C or UART master) should be connected to either the serializer or deserializer CC pins (SDA/RX, SCL/TX) at any one time within a serial link system. If both the serializer and deserializer within a single system are connected to µCs and the remote CC is not disabled, the two µCs must take turns using the CC. Without arbitrated access, concurrent transfers from each µC will result in a NACK (I2C) or a collision (UART).

Note: Multi-master UART applications have many design risks and are not recommended.

Remote-side control channel access is blocked by setting DIS_REM_CC = 1 in both devices. When remote access is disabled, GMSL2 device access is limited to locally-connected µCs, and CC access to remote-side device(s) is disabled.

A GMSL2 device can be set to read-only on the CC by setting the CFG_BLOCK register to 1. After setting a device to read-only, it cannot be configured via the CC until the device powers down (PWNDNB = 0 or V_{DD} = 0). This bit can be used to freeze device configuration.

Applications may require remote access to GMSL2 device registers while blocking I2C/UART transmissions from being repeated at the remote device's CC pins. Remote device CC pins are disabled by setting DIS_LOCAL_CC = 1 in the remote device. When disabled, these pins can be used for another function (e.g., GPIO).

Note: Some devices with reduced pin counts share main CC and pass-through functionality on the same set of pins. Extreme care must be taken when enabling and disabling the shared pins with respect to the channel being used. These devices must have DIS_LOCAL_CC = 1 set to use the pins for pass-through I2C/UART. Refer to device-specific datasheets for pinout information.

By default, main I2C/UART CC transmission on the serial link are protected by ARQ (Automatic Repeat Request) for error correction. Main I2C/UART CC ARQ and packet CRC can only be disabled by a local register write in both devices while the link is not locked. Reference the CRC Error Detection and ARQ Error Correction section for details.

Note: Disabling ARQ and packet CRC for the main I2C/UART CC is discouraged for field applications and should only be used for debugging or testing purposes.

24.2.3 Main I2C Control Channel

I2C is a bidirectional communication protocol that connects multiple devices via a single two-wire bus. Microcontrollers can be programmed to generate I2C data transfers that GMSL2 devices can process. The sections below contain configuration information.

24.2.3.1 I2C Mode Configuration

The following tables and sections contain registers associated with the main I2C control channel (CC). Note that the I2CSEL, DIS_REM_CC, DIS_LOCAL_CC, and CFG_BLOCK bitfields are common to the main I2C and UART control channel protocols.

Note: Registers listed in the following tables may not exist in all parts. Reference device-specific datasheets and register documents for the most accurate part information.

ADI Confidential for NDA Revision only
Analog Devices Confidential and Proprietary
Provided to GMSL2 Customers Under NDA
Maxim Integrated Company Confidential and Proprietary
Provided to GMSL2 Customers Under NDA

Table 91. I2C/UART Selection Status Register

Bitfield	Description	Decode
I2CSEL[0]	This bit is set according to the latched I2CSEL/CFG pin value at power-up.	0b0: UART 0b1: I2C

Note: I2CSEL[0] should only be used as a status bit, and writing to this bitfield is discouraged. It is not recommend to change the main CC from I2C to UART or vice versa by writing to this register after power-up.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Table 92. Remote/Local Control Channel and Device Configuration Registers

Bitfield	Description	Decode
DIS_REM_CC[0]	Disables the remote-control channel over the GMSL2 link.	0b0: Remote control-channel enabled 0b1: Remote control-channel disabled
DIS_LOCAL_CC[0]	Disables control-channel connection to RX/SDA and TX/SCL pins.	0b0: RX/SDA and TX/XCL connected to control channel 0b1: RX/SDA and TX/SCL disconnected from control
CFG_BLOCK[0]	Configuration block. When set, all registers become non-writable (read-only). This bit can be used to freeze the chip configuration.	0b0: Not Blocked 0b1: Blocked

Table 93. Number of I2C-to-I2C Links Configuration Registers

Bitfield	Description	Decode
I2C_AUTO_CFG[0]	When set to 1, I2C-to-I2C number of links is automatically determined based on splitter mode. In splitter mode, response from two I2C channels are expected, otherwise response from one I2C channel is expected.	0b0: Number of I2C-to-I2C links set by I2C_SRC_CNT[2:0] bits 0b1: Splitter mode automatically determines the number of I2C-to-I2C links
I2C_SRC_CNT[2:0]	I2C-to-I2C number of links (valid when I2C_AUTO_SRC = 0). Set this field to N - 1 when expecting I2C response from N remote I2C transmitters (usually the same as the number of remote devices connected to this device).	0b000: 1 serializer/deserializer connected 0b001: 2 serializers/deserializers connected 0b010: Reserved 0b011: Reserved 0b100: Reserved 0b101: Reserved 0b110: Reserved 0b111: Reserved

ADI Confidential for Newvision under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.2.3.1.1 I2C Internal Register Access

Each GMSL2 device has an internal I2C slave for register access. The internal registers can be written and read according to the I2C protocol using the data transfer formats below. Register addresses are 16-bits wide. Single or multiple data bytes can be written or read (by address auto-increment). I2C data transfers can be monitored with the read-only registers for I2C acknowledge bits and time-out status (Table 94).

Note: Devices have auto-increment limits. Reference device-specific datasheets for information.

24.2.3.1.1.1 I2C Data Transfer Formats

Note: In *Figure 67* and *Figure 68* below, unshaded blocks indicate data transfers from the master to the slave and shaded blocks indicate data transfers from the slave to the master.

I2C Write:

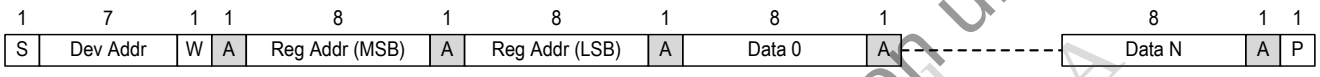


Figure 67. I2C Write Data Transfer Format

I2C Read:

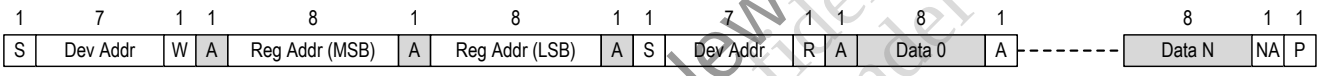


Figure 68. I2C Read Data Transfer Format

Table 94. I2C Acknowledge Bit and Time-Out Status (Read-Only Registers)

Bitfield	Description	Decode
REM_ACK_ACKED[0]	Inverse of the I2C acknowledge bit received from remote side.	0b0: I2C acknowledge bit received as 1 0b1: I2C acknowledge bit received as 0
REM_ACK_RECVED[0]	I2C acknowledge bit for any I2C byte has been received from the remote side for the previous GMSL packet with I2C data.	0b0: I2C acknowledge bit not received 0b1: I2C acknowledge bit received
I2C_TIMED_OUT[0]	Internal I2C-to-I2C slave or master has timed out while receiving data transfer from remote device.	0b0: Time-out has not occurred 0b1: Time-out has occurred

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Only for distribution to Maxim Integrated Company Confidential, Under NDA Provided to GMSL2 Customers

24.2.3.1.2 I2C over the GMSL2 Link (Remote Device Access)

The I2C channel connects the serializer and deserializer I2C interfaces (i.e., SDA_RX and SCL_TX) over the serial link.

In addition to the internal I2C slave used for register access, GMSL2 devices have I2C link masters and link slaves used for remote device access via I2C transmissions over the GMSL2 link. When an external I2C master (e.g., µC) performs an I2C transaction on the I2C bus on one side of the link (i.e., local side), the I2C events (e.g., Start, Stop, Data Bit 0, and Data Bit 1) are forwarded to the other side of the link (i.e., remote side) by the local-side device's I2C link slave. The I2C events are received by the remote-side device's I2C link master and generated on the remote-side I2C bus. The remote-side I2C link master sends back across the link any I2C events that are expected to be driven by the remote-side I2C slave(s) (e.g., Ack bits during writes and read data bits during reads) to the local-side I2C link slave.

I2C over the GMSL2 link operates such that the remote-side internal I2C master mimics the actions of the local-side external I2C master (e.g., µC) and the local-side internal I2C link slave mimics the actions of the external I2C slave(s) on the remote side. This method logically connects the two separate I2C buses: an entire I2C transaction looks like as if it has been performed on the same physical I2C bus (except for incurred timing differences).

I2C data transfers require an immediate acknowledgement from the receiver following each byte. To account for timing differences between the master and slave and to allow time for data to be forwarded and received across the serial link, the I2C protocol uses clock stretching (i.e., holding SCL low) to temporarily pause communication as the acknowledge propagates through the I2C control channel. In GMSL2 systems, all I2C devices on the local side (i.e., external master µC and any attached peripherals) must support clock stretching; the remote-side I2C peripherals are not required to support clock stretching (Figure 69).

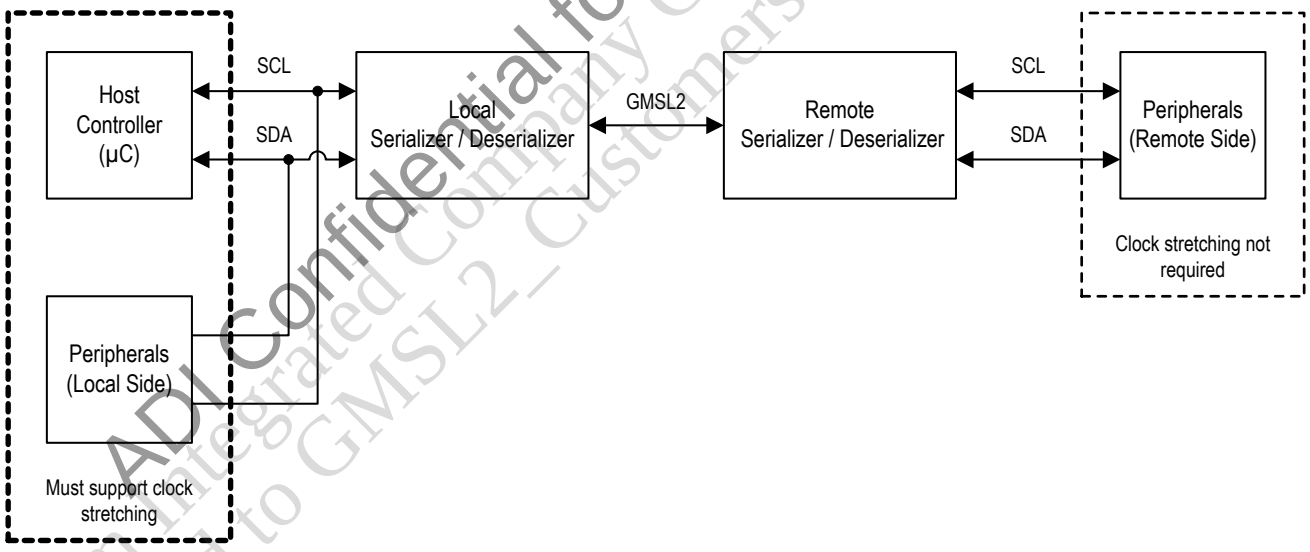


Figure 69. I2C Clock Stretching

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

The remote-side link I2C master operates according to timing settings configured with the remote-side MST_BT bitfield. It is strongly recommended that users program the MST_BT bitfield as close as possible to the bit rate used by the external master (e.g., µC). The local-side I2C slave timing is configured with the SLV_SH register. The SLV_TO and MST_TO bitfields select the time-out durations that release the local or remote side I2C bus in case an expected response from remote side is not received within the selected time-out duration. Ensure that all timing registers (Table 95 and Table 96) are programmed according to the desired I2C bit rate and in compliance with the official I2C timing parameters.

Table 95. Internal I2C Slave Configuration Registers

Bitfield	Description	Decode
SLV_SH[1:0]	I2C-to-I2C slave setup and hold time setting (setup, hold). Configures the interval between SDA and SCL transitions when driven by the internal I2C slave.	0b00: Set for I2C Fast-mode Plus speed 0b01: Set for I2C Fast-mode speed 0b10: Set for I2C Standard-mode speed 0b11: Reserved
SLV_TO[2:0]	I2C-to-I2C slave time-out setting. Internal GMSL2 I2C slave times out after the configured duration if it does not receive any response while waiting for a packet from the remote device.	0b000: 16us 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled

Table 96. Internal I2C Master Configuration Registers

Bitfield	Description	Decode
MST_BT[2:0]	I2C-to-I2C master bit rate setting. Configures the I2C bit rate used by the internal I2C master (in the device on the remote side from the external I2C master). Set this according to the I2C speed mode.	0b000: 9.92Kbps - Set for I2C Standard- mode speed 0b001: 33.2Kbps - Set for I2C Standard-mode speed 0b010: 99.2Kbps - Set for I2C Standard- or Fast-mode speed 0b011: 123Kbps - Set for I2C Fast-mode speed 0b100: 203Kbps - Set for I2C Fast-mode speed 0b101: 397Kbps - Set for I2C Fast- or Fast-mode Plus speed 0b110: 625Kbps - Set for I2C Fast-mode Plus speed 0b111: 980Kbps - Set for I2C Fast-mode Plus speed
MST_TO[2:0]	I2C-to-I2C master time-out setting. Internal GMSL2 I2C master times out after the configured duration if it does not receive any response while waiting for a packet from remote device.	0b000: 16us 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled

The external I2C master (e.g., µC) can be located on the serializer-side (typically for display applications) or the deserializer-side (typically for camera applications). Multi-microcontroller operations are supported provided that a software arbitration method is used to prevent collisions (see I2C Multi-Master Options below). The serial link assumes that only one microcontroller is transmitting at any given time.

ADI Confidential for New Version under NDA
Analog Devices Initial Distribution
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.2.3.1.3 I2C Splitter Mode

The I2C channel can be used in GMSL2 Splitter Mode applications (i.e., a single serializer connects to two deserializers). In GMSL2 splitter mode:

- The µC connected to the local device in splitter mode can communicate with each connected remote device and the attached peripherals.
- The µC connected to a remote device can communicate with the local device in splitter mode and its attached peripheral. It cannot communicate with the other remote device(s).
- If more than one µC is connected to the serial link system, only one µC can use the control channel at a time. The serial link assumes that only one µC is using the CC at any one time. Multiple µCs must take turns using the control channel to avoid error conditions. If the CC is in use by a µC, attempts from another µC to access the CC will result in a NACK. If software arbitration is not implemented, the remote-side controller should be blocked to avoid collisions.

Note: The I2C channel can also operate in GMSL2 *Reverse Splitter Mode*. The operation is the same as splitter mode but reversed.

I2C address configuration options are presented in Table 97 below. Each function is described in more detail in following sections. Note that these functions may be used in combination.

Table 97. I2C Address Configuration

Feature	Description	Purpose	Notes
I2C Address Reassignment	Allows users to configure a GMSL2 device's I2C address from the default address at power-up (determined by the CFG pin).	Used to assign unique I2C addresses to GMSL2 devices when a GMSL2 system comprises multiple devices with the same default address.	In splitter and reverse splitter applications, I2C address reassignment is used to avoid address conflicts and data collisions.
I2C Address Translation	Configurable mapping of one I2C address to another (virtual) I2C address. Address translation occurs within the GMSL2 device.	Map a virtual I2C address to a peripheral I2C device on the remote side of the GMSL2 link. This allows a host µC to separately access different devices and peripherals with the same device address via a unique virtual I2C address.	The mapped virtual I2C address must be unique within the system.
I2C Broadcasting	Allows multiple GMSL2 devices to be addressed simultaneously at a single I2C address.	Simplifies configuration by allowing a single write to simultaneously configure multiple devices as a group to identical settings.	It is not possible to determine if all devices provide an acknowledge following an I2C broadcast write. Each device must be individually checked to verify correct configuration.

ADI Confidential for New Vision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.2.3.1.4 I2C Address Reassignment

Note: The methodology discussed in this section also applies to UART.

In splitter and reverse splitter mode applications, it is recommended to use a unique I2C device address for each connected serializer/deserializer. I2C address reassignment is required if identical device addresses are selected at power-up.

Note: The follow configuration procedures do not apply to GMSL2 quad deserializer devices. Refer to the Quad Deserializer section for more information.

24.2.3.1.4.1 Display Setup – One Serializer to Two Deserializers

In systems with two deserializers connected to one serializer, address reassignment should be used to avoid address/data conflicts.

The procedure below applies to a system comprising two identical deserializers connected to one display serializer, with the microcontroller located on the serializer side.

1. Isolate one deserializer by configuring the serializer into single link mode. For example, enable link A only by setting LINK_CFG[1:0] to 0b01 and setting AUTO_LINK[0] low on the serializer.
2. Perform a link reset by writing the self-clearing bit RESET_ONESHOT high.
3. Poll the LOCKED pin until it goes high.
4. Modify the deserializer I2C device address connected to link A with a register write to DEV_ADDR[6:0] located in REG0.
5. Modify each of the source identifiers (TX_SRC_ID) for each of the GMSL protocol packets to a value unique relative to the other deserializer in the system. See Table 98 for an example of the default TR3 register values of each packet type based on the MAX96754 power-on I2C addresses.
6. Configure the serializer to splitter mode by writing LINK_CFG[1:0] to 0b11.
7. Perform a link reset by writing the self-clearing bit RESET_ONESHOT high.
8. Poll the LOCKED pin until it goes high.
9. All devices should be present on the I2C bus. Continue with any additional system configuration.

Table 98. Default Values of TR3 Registers for MAX96754

Default Register Values (Hex)	Register Address	Packet Type	I2C Addresses (Default Values)							
			0x90	0x94	0x98	0xD0	0xD4	0xD8	0x50	0x54
	0x5B	Audio	30	31	32	33	34	35	36	37
	0x63	Info-Frame	30	31	32	33	34	35	36	37
	0x6B	SPI	30	31	32	33	34	35	36	37
	0x73	I2C/UART	30	31	32	33	34	35	36	37
	0x7B	GPIO	30	31	32	33	34	35	36	37
	0x8B	AHDCP	30	31	32	33	34	35	36	37
	0xA3	Pass Through 1	30	31	32	33	34	35	36	37
	0xAB	Pass Through 2	30	31	32	33	34	35	36	37

ADI Confidential for Review Only
Maxim Integrated Confidential, Provided to GMSL2 Customer Under NDA

24.2.3.1.4.2 Camera Setup – Two Serializers to One Deserializer

In systems with two camera modules connected to one deserializer, address reassignment should be used to avoid address conflicts and data collisions.

The procedure below applies to a system comprising two identical camera modules connected to a dual CSI-2 camera deserializer, with the microcontroller on the deserializer side.

1. Isolate one camera module by configuring the deserializer into single link mode. For example, enable link A only by setting LINK_CFG[1:0] to 0b01 and AUTO_LINK[0] low on the deserializer.
2. Perform a link reset by writing the self-clearing bit RESET_ONESHOT high.
3. Poll the LOCKED pin until it goes high.
4. Modify the serializer I2C device address connected to link A with a register write to DEV_ADDR[6:0] located in REG0.
5. Modify each of the source identifiers (TX_SRC_ID) for each of the GMSL protocol packets to a value unique relative to the other serializer in the system. See Table 99 for an example of the default TR3 register values of each packet type based on the MAX9295E power-on I2C addresses.
6. Configure the deserializer to reverse splitter mode by writing LINK_CFG[1:0] to 0b11.
7. Perform a link reset by writing the self-clearing bit RESET_ONESHOT high.
8. Poll the LOCKED pin until it goes high.
9. All devices should be present on the I2C bus. Continue with any additional system configuration.

Table 99. Default Values of TR3 Registers for MAX9295E

	Register Address	Packet Type	I2C Addresses (Default Values)			
			0x80	0x84	0xC0	0xC4
Default Register Values (Hex)	0x6B	Audio	10	10	12	12
	0x73	Audio	11	11	13	13
	0x7B	Info-Frame	30	31	32	33
	0x83	SPI	30	31	32	33
	0x8B	I2C/UART	30	31	32	33
	0x93	GPIO	30	31	32	33
	0xA3	Pass Through 1	30	31	32	33
	0xAB	Pass Through 2	30	31	32	33

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution Prohibited
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2_Customers

24.2.3.1.5 I2C Address Translation

Address translation is a function in I2C mode that enables mapping one device address to a virtual device address. In GMSL2 serial link systems, two separate device addresses can be translated to another two separate device address. This function is used when two identical modules with the same device address are used within the same system. Address translation allows the host µC to separately access different devices and peripherals with the same device address via software configuration. Configuration details are presented in Table 100.

Note: There may be more than two devices/modules with the same address used with GMSL2 Quad Deserializer systems. Reference the CSI-2 Quad Deserializer section for additional information.

Table 100. I2C Address Translator Configuration Registers

Bitfield	Description	Decode
SRC_A[6:0]	I2C address translator source A. When I2C device address matches I2C SRC_A, internal I2C master (on remote side) replaces the device address by I2C DST_A.	0bXXXXXXX: Value of I2C SRC_A
DST_A[6:0]	I2C address translator destination A. See the description of I2C SRC_A.	0bXXXXXXX: Value of I2C DST_A
SRC_B[6:0]	I2C address translator source B. When I2C device address matches I2C_SRC_B, internal I2C master (on remote side) replaces the device address by I2C_DST_B.	0bXXXXXXX: Value of I2C SRC_B
DST_B[6:0]	I2C address translator destination B. See the description of I2C SRC_B.	0bXXXXXXX: Value of I2C DST_B

Address translation can be used in addition to address reassignment (see Camera Setup – Two Serializers to One Deserializer) for systems with two identical camera modules connected to one deserializer. Address translation allows two camera modules with image sensors at the same I2C address to be independently addressed at user-defined virtual addresses to avoid address conflicts. GMSL2 devices allow for up to two address translations (permitting two devices to have their addresses translated). In this scenario, I2C address translation is configured for the serializer device. I2C commands are sent from the deserializer over the serial link to the serializer. In the serializer, address SRC_A[6:0] is then translated into address DST_A[6:0] by the remote link master (Table 100). Both the serializer and the peripheral(s) will see this translated address (i.e., the DST_A address).

Note: An unused address translation can be used for I2C broadcasting. Refer to the *I2C Broadcasting* section for details.

ADI Confidential Pre-Release Version under NDA
Analog Devices Confidential, Subject to Maxim Integrated Non-Disclosure Agreement
Provided to GMSL2_CustomerUnder NDA

The procedure below provides the configuration steps for two identical camera modules connected to a CSI-2 deserializer. The µC is located on the deserializer side (Figure 70).

- Use address reassignment to modify one serializer's I2C device address so that each serializer has a unique address. Refer to address reassignment procedure given above (see Camera Setup – Two Serializers to One Deserializer). Ensure that the deserializer is in reverse splitter mode and that all devices are present on the I2C bus before proceeding with the following steps.
- Program the address translation registers in the (link A) serializer by setting the desired image sensor address in the source register SRC_A[6:0] and the original image sensor address in the destination register DST_A[6:0].
- Program the address translation registers in the (link B) serializer by setting the desired image sensor address in the source register SRC_A[6:0] and the original image sensor address in the destination register DST_A[6:0].

24.2.3.1.5.1 I2C Address Translation Example

An I2C address reassignment and translation example is shown in the block diagram below (*Figure 70*). A GMSL2 CSI-2 deserializer (MAX9296A) is connected to two identical camera modules. Each camera module comprises a GMSL2 serializer (MAX9295A) at I2C address 0x80 and an image sensor at address 0x6C.

I2C address reassignment is used to change the link A serializer's device address from 0x80 to 0x84. Then, I2C address translation is performed on each serializer to allow individual access of each image sensor. The I2C address for the link A serializer is translated from 0x20 to 0x6C; the I2C address for the link B serializer is translated from 0x22 to 0x6C.

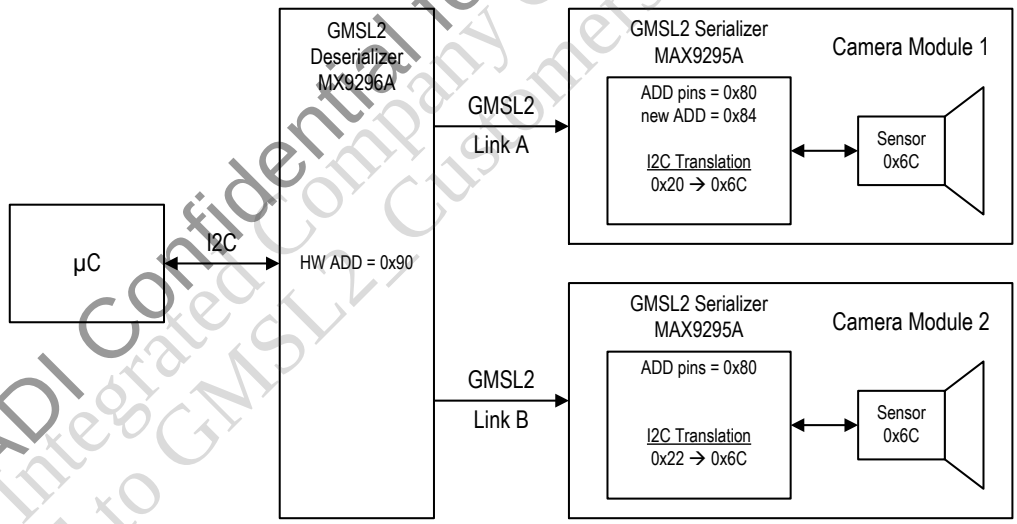


Figure 70. I2C Address Translation

Analog Devices Confidential for New Vision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA
Initial Distribution Only

Programming sequence of this example:

```
# Enable link A only on the MAX9296A to isolate Camera Module 1
0x90,0x10,0x01
# Apply reset one shot
0x90,0x10,0x21
# Poll the LOCKED pin until it goes high or add enough delay for the link to lock
# Use I2C address reassignment for the MAX9295A connected to link A to change
device address from 0x80 to 0x84
0x80,0x00,0x84
# Modify each of the source identifiers (TX_SRC_ID) for each of the GMSL protocol
packets on the MAX9295A connected to link A. Note that different devices can have
different GMSL protocol packets. Reference device-specific datasheets and register
documents.
0x84,0x7B,0x11
0x84,0x83,0x11
0x84,0x8B,0x11
0x84,0x93,0x11
0x84,0xA3,0x11
0x84,0xAB,0x11
# Configure the MAX9296A to reverse splitter mode
0x90,0x10,0x03
# Apply reset one shot
0x90,0x10,0x23
# Poll the LOCKED pin until it goes high or add enough delay for the link to lock
# Program the address translation registers in the (link A) serializer
0x84,0x42,0x20
0x84,0x43,0x6C
# Program the address translation registers in the (link B) serializer
0x80,0x42,0x22
0x80,0x43,0x6C
```

ADI Confidential for NewVision under NDA

Analog Devices Confidential, distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

24.2.3.1.6 I2C Broadcasting

I2C broadcasting can be used to simplify programming in GMSL2 systems consisting of a deserializer connected to two identical camera modules. I2C broadcasting enables the deserializer to configure the connected camera modules as a group to identical settings. The deserializer uses the I2C address of the image sensors assigned at power-up and the I2C address of the serializers assigned via I2C address translation as the broadcasting address.

Additional configuration may be necessary depending on the system application. If individual configuration of the image sensors is also required, each must be given a unique address with I2C Address Translation. This allows the deserializer to access the image sensors both as a group via I2C broadcasting and individually with the unique addresses.

Note: To assign unique addresses to each image sensor, an I2C address translation must be performed on each device.

To avoid address conflicts and data collisions, use I2C Address Reassignment to assign a unique I2C address to each serializer (reference Camera Setup – Two Serializers to One Deserializer). If serializers with unique addresses share many configuration settings, address translation can be used to assign a broadcasting address so that the serializers can be configured as a group to identical settings by the deserializer. Note that programming a broadcasting address for both serializers requires one address translation per serializer.

Note: ADI cannot guarantee the programming of all devices. Each device must be individually checked to verify correct configuration, as ADI cannot guarantee or distinguish which device provides the acknowledge. If one slave pulls the SDA line low, it can mask the state of the other slaves. ADI cannot guarantee that all slaves pulled SDA low. Therefore, the user must implement a mechanism to verify that each slave is properly configured.

The procedure below provides the configuration steps for two identical camera modules connected to a CSI-2 deserializer. The µC is located on the deserializer side (Figure 71).

1. Use address reassignment to modify one serializer's I2C device address to have a unique address for each serializer. Refer to the I2C Address Reassignment procedure given above (see Camera Setup – Two Serializers to One Deserializer). Ensure that the deserializer is in reverse splitter mode and that all devices are present on the I2C bus before proceeding with the following steps.
2. Modify the first address translation register in the (link A) serializer to give a broadcast address to the serializer. Set the desired broadcasting address in the source register SRC_A[6:0] and the modified serializer address at step 1 above in the destination register DST_A[6:0].
3. Modify the second translation register in the (link A) serializer to give a unique address to the image sensor. Set the desired image sensor address in the second source register SRC_B[6:0] and the original image sensor address in the second destination register DST_B[6:0].
4. Modify the first address translation register in the (link B) serializer to give a broadcast address to the serializer. Set the desired broadcasting address in the source register SRC_A[6:0] and the original serializer address in the destination register DST_A[6:0].
5. Modify the second translation register in the (link B) serializer to give a unique address to the image sensor. Set the desired image sensor address in the second source register SRC_B[6:0] and the original image sensor address in the second destination register DST_B[6:0].

ADI Confidential for NEWVision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to CHIPSEA Employees Under NDA

24.2.3.1.6.1 I2C Broadcasting Example

An example of I2C address reassignment, translation, and broadcasting is shown in the block diagram below (*Figure 71*). A GMSL2 CSI-2 deserializer (MAX9296A) is connected to two identical camera modules. Each camera module comprises a GMSL2 serializer (MAX9295A) at I2C address 0x80 and an image sensor at address 0x6C.

I2C address reassignment is used to change the link A serializer's device address from 0x80 to 0x84. Then, I2C address translation is performed twice on each serializer to allow individual access of each image sensor and configure an I2C broadcasting address for the serializers. For individual access of the image sensors, the I2C address for the link A serializer is translated from 0x20 to 0x6C, and the I2C address for the link B serializer is translated from 0x22 to 0x6C. The serializers have an additional I2C translation programmed to establish the I2C broadcasting address. Here, the broadcasting address is set as the source register, and the serializer device address is set as the destination register. In the link A serializer, 0xC4 is translated to 0x84 (i.e., the modified device address configured with I2C address reassignment); in the link B serializer, 0xC4 is translated to 0x80.

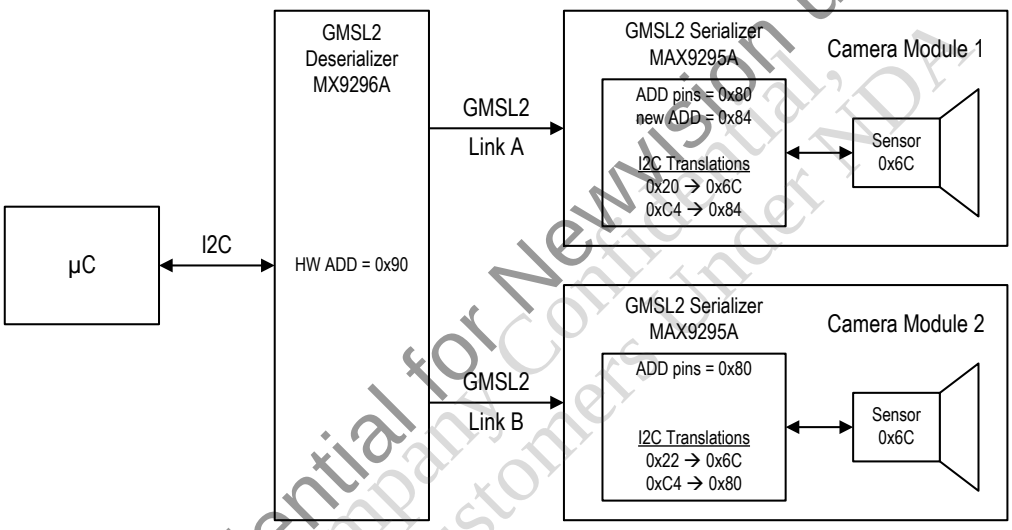


Figure 71. I2C Broadcasting

ADI Confidential for New Version under NDA
Analog Devices Maxim Integrated Company Confidential, Provided to GMSL2_Customers_Under_NDA

Programming sequence of this example:

```
# Enable link A only on the MAX9296A to isolate Camera Module 1
0x90,0x10,0x01
# Apply reset one shot
0x90,0x10,0x21
# Poll the LOCKED pin until it goes high or add enough delay for the link to lock
# Use I2C address reassignment for the MAX9295A connected to link A to change
device address from 0x80 to 0x84
0x80,0x00,0x84
# Modify each of the source identifiers (TX_SRC_ID) for each of the GMSL protocol
packets on the MAX9295A connected to link A. Note that different devices can have
different GMSL protocol packets. Reference device-specific datasheets and register
documents.
0x84,0x7B,0x11
0x84,0x83,0x11
0x84,0x8B,0x11
0x84,0x93,0x11
0x84,0xA3,0x11
0x84,0xAB,0x11
# Configure the MAX9296A to reverse splitter mode
0x90,0x10,0x03
# Apply reset one shot
0x90,0x10,0x23
# Poll the LOCKED pin until it goes high or add enough delay for the link to lock
# Program the 1st address translation registers in the (link A) serializer to give a
broadcast address to the serializer
0x84,0x42,0xC4
0x84,0x43,0x84
# Program the 2nd address translation registers in the (link A) serializer to give a
unique address to the image sensor
0x84,0x44,0x20
0x84,0x45,0x6C
# Program the 1st address translation registers in the (link B) serializer to give a
broadcast address to the serializer
0x80,0x42,0xC4
0x80,0x43,0x80
# Program the 2nd address translation registers in the (link B) serializer to give a
unique address to the image sensor
0x80,0x44,0x22
0x80,0x45,0x6C
```

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

24.2.3.2 I2C Multi-Master Options

GMSL2 systems can support multi-master I2C applications. However, measures to avoid conflicts/collisions must be implemented to ensure proper operation. There is no officially supported measure; options are provided below.

- **Islanding**: The remote CC can be disabled to separate the two µCs. Refer to Disabling Remote Control Channel on Power-up for details.
- **Pass-through Channel**: Depending on the application and its requirements, one of the pass-through I2C channels can be used to separate the two I2C masters.
- **Single Master**: One of the µCs is designated as the master. Only the master sends I2C commands through the GMSL2 link. Alternatively, software arbitration can permit a second master to send I2C commands (after a delay) when the first master is not active.
- **Token Pass**: Similar to single master, except that the master also sets up the GPIOs. The slave can request control through one of the GPIOs, and the master can notify the slave that control is allowed through a different GPIO. Note that most GPIOs are low by default, and polarity must be carefully chosen to ensure that the slave µC waits if the GPIOs are not yet programmed.

If one of the µCs does not need to program GMSL2 devices, the pass-through I2C channel should be used instead of the I2C CC.

Note: Multi-master UART applications have many design risks and are not recommended.

24.2.3.3 I2C Channel GMSL2 Bandwidth Utilization

Refer to the GMSL2 Link Bandwidth Consumption from Side Channels – I2C section for details.

24.2.3.4 I2C Debug Techniques

The I2C protocol is used by most µCs. Ensure that µCs are programmed to generate I2C data transfers that GMSL2 devices can process and respond to I2C data transfers from GMSL2 devices.

If I2C issues are exhibited, inspect the I2C port(s) using a logic analyzer or digital oscilloscope. Verify that the waveforms observed on the SDA and SCL pins are as expected.

The REM_ACK_RECVED register is used to check if an I2C Ack bit for any I2C byte has been received from the remote side for the previous I2C data transfer. Alternatively, a logic analyzer or digital oscilloscope can be used to monitor if I2C Ack bits have been received from the remote side for any transmitted I2C byte.

The REM_ACK_ACKED register can be read back to see if the received Ack is 1 or 0.

The I2C_TIMED_OUT register can be read back to check if the internal I2C–I2C link slave or link master has timed-out while receiving data from the remote device.

ADI Confidential for New vision under NDA
Analog Devices Confidential, Subject to Non-Disclosure Agreement
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

24.2.4 Main UART Control Channel

UART is typically a point-to-point communication protocol; however, in GMSL2 systems, it is possible to have microcontroller(s), serializer(s), deserializer(s), and peripheral slave(s) physically connected to the same bus. This expanded application of UART involves several important restrictions with ramifications on system design.

24.2.4.1 UART Mode Configuration

The following tables and sections contain registers that are associated with the main UART control channel. Note that the I2CSEL, DIS_REM_CC, DIS_LOCAL_CC, and CFG_BLOCK registers are common to the main I2C and UART control channel protocols. Reference the I2C Mode Configuration section for information regarding the shared registers (Table 91 and Table 92).

Devices with multiple independent GMSL2 links require that UART programming be performed on a link-by-link basis because UART communication is not broadcast across all links. Program the UART_0_LINK_SELECT, UART_1_LINK_SELECT and UART_2_LINK_SELECT registers accordingly. This contrasts with I2C, which can be used to communicate over any number of available links simultaneously.

Note: Registers listed in the following sections and tables may not exist in all parts. Reference device-specific datasheets and register documents for the most accurate part information.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2_Customers Under NDA

24.2.4.1.1 UART Base/Bypass Mode

GMSL2 devices provide two main UART CC modes of operation: UART Base Mode and UART Bypass Mode. Base Mode is used to program and configure the serializer and deserializer. The µC can communicate with both GMSL2 devices and attached (compatible) peripherals on the remote side. Bypass Mode is used to bypass GMSL2 devices and provides direct point-to-point access to a peripheral device. The UART CC can be programmed to transition between these modes, allowing flexible use of the UART CC.

UART Bypass Mode is distinct from Pass-through UART: Pass-through UART only provides access to remote peripherals (internal GMSL2 device registers are inaccessible), while UART Bypass Mode is a control channel mode. If a GMSL2 system includes a peripheral connected to the remote device that is not compatible with the GMSL2 UART protocol, UART Bypass Mode allows a µC to communicate with this peripheral without requiring the use of the UART Pass-through connection. After the communication is complete, the µC can access GMSL2 devices in UART Base Mode. Pass-through UART does not have access to GMSL2 devices.

24.2.4.1.1.1 UART Base Mode

UART Base Mode is enabled by default at power-up. In base mode, µCs are the host and use the GMSL2 *UART Frame Format* to write and read the internal registers of GMSL2 devices from either side of the serial link. The µC can also communicate with attached remote peripherals compatible with the GMSL2 UART protocol. UART data transmitted by the µC is output from the remote-side GMSL2 device Tx pin by default. To disable the UART Tx and Rx pins on remote-side GMSL2 device, set DIS_LOCAL_CC = 1 in the remote GMSL2 device.

Single or multiple bytes can be written or read using the GMSL2 UART protocol. Between each UART transmission, the µC must wait up to 48-bits time or 200µs, whichever is longer, for the expected response from the GMSL2 device. This logically enforces half-duplex control channel operation. After completely receiving the response to a transmission, the µC must wait and keep the line high for at least 48-bits times before sending the next transmission. The high-time duration sums up to 49 bits with the stop bit of previous data transfer included. Note that response to a read or write request (including Ack frame) is also part of the data transfer, and the 48-bits time wait starts after the stop bit of the last UART frame of the response.

The first received UART byte of a transmission (i.e., sync or acknowledge frame) can be configured to be delayed by 0-, 1-, 4-, or 8-bits time with the OUT_DELAY register (Table 101) in UART base mode. This programming can be used to ensure that UART frames of the same data transfer are output one after the other on the remote side.

Table 101. UART Initial Output Delay Configuration Register

Bitfield	Description	Decode
OUT_DELAY[1:0]	UART initial output delay. In base mode, the first received UART byte of a packet (sync or acknowledge frame) is delayed by the configured number of bit times in order to output the UART frames of the same packet back-to-back on remote side.	0b00: 0 bits 0b01: 4 bits 0b10: 8 bits 0b11: 1 bit

ADI Confidential for Review Only
Maxim Integrated GMSL2 Customers Under NDA
Provided to GMSL2 Customers Under NDA

24.2.4.1.1.2 UART Bypass Mode

In UART bypass mode, UART commands are not interpreted by the GMSL2 devices, and data is passed directly from the µC to connected peripherals (and from peripherals to the µC). The µC cannot access GMSL2 device registers. There are two methods to enable UART bypass mode: with the BYPASS_EN register bit or with the MS pin (shared with GPIO). Configure the MS pin method of UART bypass mode enable with REM_MS_EN and LOC_MS_EN (Table 102).

Note: Ensure that the GPIO pins with MS pin assignments are available for use before configuring the system for MS pin control of UART bypass mode. Reference device-specific datasheet for pinout information.

Note: UART bypass mode should ONLY be enabled when GMSL2 link lock is preset.

UART bypass mode is implemented with software by setting the BYPASS_EN register in the remote device first, then in the local device. This programming can be temporary or permanent. In temporary programming, bypass mode is automatically exited and the BYPASS_EN register is reset to 0 when the UART line stays high beyond the time-out duration defined with the BYPASS_TO register. BYPASS_TO has four settings: 2ms, 8ms, 32ms, and "Disabled". Permanent programming is configured by setting BYPASS_TO to "Disabled": BYPASS_EN is never cleared, and the device stays in bypass mode until power is cycled. UART bypass mode configuration registers are presented in Table 102.

Note: Programming BYPASS_TO to "Disabled" is discouraged because this setting blocks control channel access to the GMSL2 devices.

Table 102. UART Bypass Mode Configuration Registers

Bitfield	Description	Decode
REM_MS_EN[0]	Enables UART bypass mode control by remote GPIO pin. When set, remote chip's GPIO is used as MS pin (UART Mode Select). Refer to the specific device's datasheet to see which GPIO has MS functionality. When MS is high, chip is in bypass mode, otherwise chip is in base mode.	0b0: UART bypass mode not controlled by remote MS pin 0b1: UART bypass mode controlled by remote MS pin
LOC_MS_EN[0]	Enables UART bypass mode control by local GPIO pin. Set to use relevant GPIO pin as MS pin (UART Mode Select). Refer to device datasheet or UART Base/Bypass Mode Operation section below to see which GPIO has MS functionality. When MS is high, chip is in bypass mode, otherwise chip is in base mode.	0b0: UART bypass mode not controlled by local MS pin 0b1: UART bypass mode controlled by local MS pin
BYPASS_DIS_PAR[0]	Selects whether or not to receive and send parity bit in bypass mode.	0b0: Receive and send parity bit in bypass mode 0b1: Do not receive and send parity bit in bypass mode
BYPASS_TO[1:0]	UART soft bypass time-out duration. When set to 0b11, BYPASS_EN is never cleared, so the device stays in bypass mode until next power down.	0b00: 2ms 0b01: 8ms 0b10: 32ms 0b11: Disabled
BYPASS_EN[0]	Enables UART soft bypass mode. Bypass mode remains active if there is UART activity. When there is no UART activity for selected duration configured by	0b0: UART soft bypass mode disabled

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential
Provided to EMS Customers Under NDA

	BYPASS_TO register, device exits bypass mode, and the bit is automatically cleared.	0b1: UART soft bypass mode enabled

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

The procedure below provides the steps required to control UART bypass mode with the MS pin.

Note: Refer to device-specific datasheet or Table 103. In the procedure below, the GPIO number corresponding to the MS pin is represented by *.

- Drive the MS pin low.
- If the remote device is an OLDI deserializer or an eDP/DP deserializer:
  - Set GPIO_TX_ID_* of the local device to 2.
- Else (remote device is not an OLDI deserializer or an eDP/DP deserializer):
  - Set GPIO_TX_ID_* of the local device to 29.
- Set GPIO_TX_EN_* = 1 in the local device.
- Set REM_MS_EN = 1 in the remote device.
- Set LOC_MS_EN = 1 in the local device.
- Repeat the following as needed:
  - Drive MS pin high from local side to switch to bypass mode.
  - Perform UART communication with the peripheral UART slave.
  - Drive MS pin low from local side to switch to base mode.
  - Perform UART communication with serializer or deserializer.

UART is in bypass mode when MS is high; UART is in base mode when MS is low.

Table 103. GPIO ID for MS Functionality

Device Family	GPIO ID for MS Function
HDMI Serializers	2
Advanced HDMI Serializers	2
DSI Serializers	0
Advanced DSI Serializers	0
CSI-2 Serializers	8
Advanced CSI-2 Serializers **MAX9295D** **MAX9295E**	 10 0
CSI-2 Camera Deserializers	0
CSI-2 Display Deserializers	0
OLDI Deserializers	2
eDP/DP Deserializers	2
CSI-2 Quad Deserializers	0

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

24.2.4.1.2 UART Splitter Mode

The UART channel can be used in GMSL2 Splitter Mode applications (i.e., a single serializer connected to two deserializers). In GMSL2 splitter mode:

- The µC connected to the local device in splitter mode can communicate with each connected remote device and the attached peripheral(s).
- The µC connected to a remote device can communicate with the local device in splitter mode and its attached peripheral(s). It cannot communicate with the other remote device(s) or attached peripheral(s).
- If a serial link system in splitter mode has multiple remote devices connected to µCs, the first µC to communicate with the local device in splitter mode gets dedicated access to the UART link (the UART link from the other µC is blocked at the splitter device). The UART link remains dedicated to that µC if it maintains communication. When the µC stops communication for the UART arbitration time-out duration (default = 2ms, programmable by ARB_TO_LEN), the link becomes available again and the process repeats (Table 104). Any µC can attempt to communicate on the UART link until it is made available and is able to gain dedicated access.

Note: The UART channel can also operate in GMSL2 *Reverse Splitter Mode*. The operation is the same as splitter mode but reversed.

Reference the I2C Splitter Mode section for shared configuration details.

Table 104. UART Rx Source Arbitration Time-Out Configuration Register

Bitfield	Description	Decode
ARB_TO_LEN[1:0]	UART RX source arbitration time-out duration. UART RX processes packets from a single UART source at any time. When UART RX does not receive any UART packets for this duration, it will select the next UART source according to the source ID of the next following received packet.	0b00: 1ms 0b01: 2ms 0b10: 8ms 0b11: 32ms

ADI Confidential for New Vision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.2.4.2 UART Frame Format

A regular UART frame with an even parity bit is used to carry one byte of data (Figure 72). UART frames consist of a low start bit, 8 data bits, a parity bit, and a high stop bit. The parity bit is high if the number of ones in 8-bit data is odd, otherwise it is low. There must be at least one high stop bit. If the next frame is in the same transmission, there can be at most four high bits from the end of the stop bit to the beginning of the next start bit. If there is a parity bit error, the transmission is discarded starting from the frame with the error.

The phase of the internal UART bit clock is adjusted using the start bit of each frame. The UART receiver asynchronously samples the incoming data at a higher rate than the actual data rate and establishes the period of the data such that it can correctly identify each bit (i.e., the incoming data is quantized).

Note: UART rates up to 5Mbps are given sufficient bit-error protection from the internal clock recovery running at 150MHz (30x oversampling).

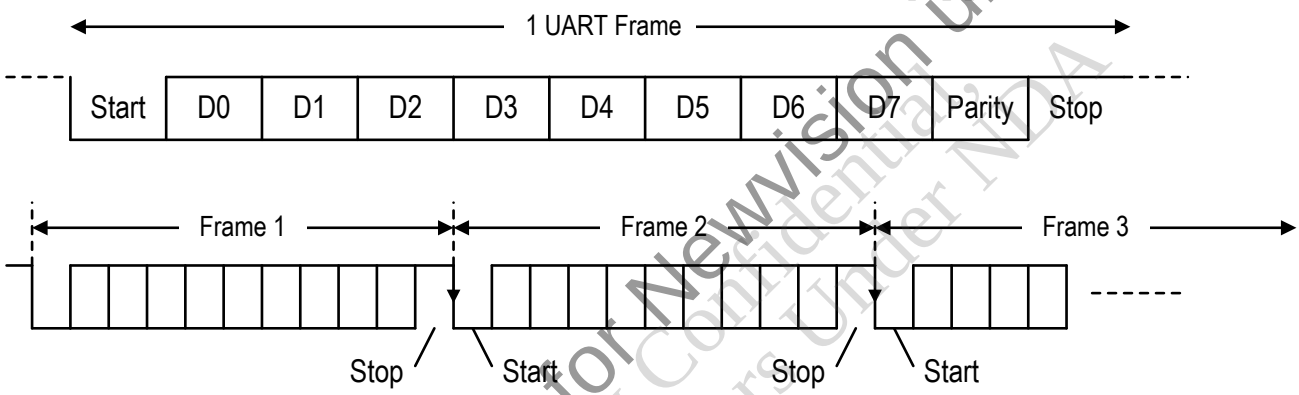


Figure 72. UART Frame Format

In bypass mode, the parity bit is enabled by default, but GMSL2 devices do not check frames for correct parity. Even or odd parity can be used. The parity bit is passed through the serial link along with the UART data so that the receiver can check for errors. The parity bit can be disabled in bypass mode by setting BYPASS_DIS_PAR to 1 before entering bypass mode.

The bit rate (baud rate) in bypass mode must be same as the last bit rate used in base mode.

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.2.4.2.1 UART Synchronization Frame

The UART bit rate is unknown to GMSL2 devices; the internal bit-length counters must be calibrated to correctly recover UART frames. GMSL2 devices use UART sync frames to calibrate the bit length in terms of the internal 150MHz clock. Each UART transmission begins with a sync frame (Figure 73). Sync frames are regular UART frames with a value of 0x79. They must be successfully detected to ensure that the remaining frames of the transmission are received correctly, as the data pattern of the sync frame is used to set the UART data bit rate for the rest of the data transfer. No transactions are allowed (the line must stay high) for a minimum of 48-bits time between UART transmissions.

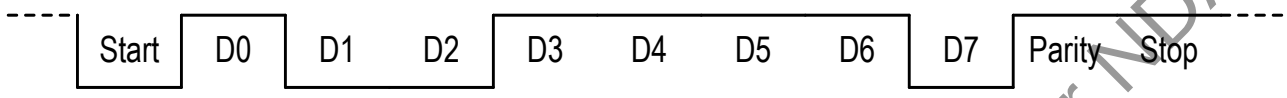


Figure 73. UART Synchronization Frame

24.2.4.2.2 UART Write Protocol

The UART write protocol consists of a 5-byte header followed by one or more data bytes (Figure 74). The LSB of the device address frame is 0. The addressed device responds with an Ack frame if no errors are detected and the transmission is valid. The byte count indicates the number of data bytes to be written (N). This must be a non-zero number.

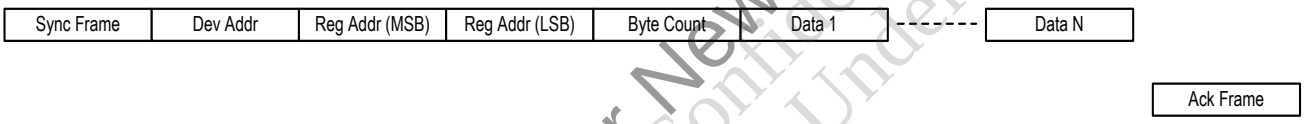


Figure 74. UART Write Protocol Format

24.2.4.2.3 UART Read Protocol

The UART read protocol consists of 5 bytes (Figure 75). The LSB of the device address frame is 1. The addressed device responds with an Ack frame followed by one or more data bytes if no errors are detected and the transmission is valid. The byte count indicates the number of data bytes to be read (N). This must be a non-zero number.

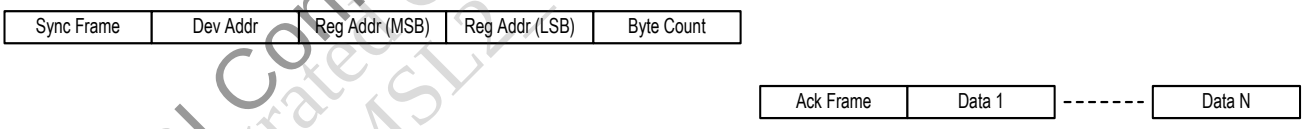


Figure 75. UART Read Protocol Format

ADI Confidential for Newvision under NDA
Analog Devices Confidential Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

24.2.4.2.4 UART Acknowledge Frame

The Ack frame is a regular UART frame with a value of 0xC3 (Figure 76). When a transmission is successfully received and recognized, the addressed device responds with an acknowledge (Ack) frame to inform the µC that the transmitted data was received, no errors were detected, and it was recognized as valid. The addressed device responds with an Ack frame after the last bit of the transmission is received.

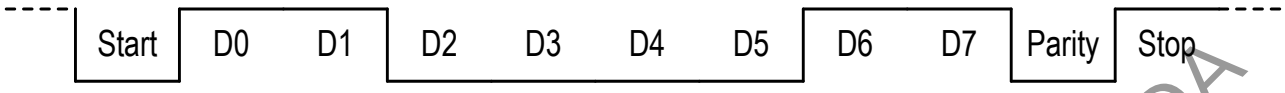


Figure 76. UART Acknowledge Frame Format

24.2.4.2.5 UART Bit Rate

In base mode, GMSL2 devices automatically detect the UART bit rate using the sync frame at the start of each transmission. The UART bit rate can be any value from 9.6Kbps to 1Mbps and can be changed (by the µC) after a transaction is completed (i.e., when the µC receives an Ack for a write or Ack and data for a read). When changing to a lower bit rate, the ratio of high and low bit rates must not exceed a factor of 3.5. The µC can begin transmissions with the new bit rate after waiting 48-bits time (measured by the slower of the old and new bit rates).

In bypass mode, the UART bit rate cannot be changed. The last bit rate used in base mode before entering bypass mode remains the bit rate for bypass mode.

ADI Confidential for New vision under NDA
Analog Devices Confidential Distribution Allowed
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.2.4.3 UART Channel GMSL2 Bandwidth Utilization

Refer to the GMSL2 Link Bandwidth Consumption from Side Channels –UART section for details.

24.2.4.4 UART Debug Techniques

If UART issues are exhibited, inspect the UART port(s) using a logic analyzer or digital oscilloscope. Verify that the waveforms observed on the UART Tx and Rx pins are as expected.

BITLEN_LSB and BITLEN_MSB are read-only bits that contain the UART bit rate detected by the device (Table 105). The value in these status bits is the ratio of 150MHz to the detected UART bit rate (e.g., it is ~300 when UART bit rate is 500Kbps).

Table 105. UART Detected Bit Length (Read-Only Registers)

Bitfield	Description	Decode
BITLEN_LSB[7:0]	UART detected bit length in terms of internal 150MHz clock, low 8 bits.	0xXXXXXXXX: UART detected bit length, low 8 bits
BITLEN_MSB[5:0]	UART detected bit length in terms of internal 150MHz clock, high 6 bits.	0bXXXXXX: UART detected bit length, high 6 bits

UART_RX_OVERFLOW and UART_TX_OVERFLOW are read-only bits that latch if an overflow condition has occurred with UART CC communications across the serial link (Table 106).

Note: In rare circumstances, a link reset during a UART communication or insufficient serial link bandwidth may result in overflow conditions.

Table 106. UART Rx/Tx FIFO Overflow Status (Read-Only Registers)

Bitfield	Description	Decode
UART_RX_OVERFLOW[0]	UART RX FIFO overflow. Set to 1 following an overflow condition. Clears upon read.	0b0: No overflow occurred 0b1: Overflow occurred
UART_TX_OVERFLOW[0]	UART TX FIFO overflow. Set to 1 following an overflow condition. Clears upon read.	0b0: No overflow occurred 0b1: Overflow occurred

ADI Confidential for New Vision Under NDA
Analog Devices Confidential Property
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

24.2.5 Disabling Remote Control Channel on Power-up

The method below describes the process to power up a GMSL2 link with the control channel disabled. This is important when a link is to be established without any I2C/UART communication visible on the remote device.

1. Keep PWDNB low initially at local device (or keep VDD off).
2. Set PWDNB high to power up the GMSL2 device.
3. Wait ~2ms for power-up to complete.
4. Write RESET_LINK high within 10ms (before the LOCKED status bit goes high) to prevent the serializer and deserializer from locking.
5. Set DIS_REM_CC high.
6. Write RESET_LINK low.
7. Poll the LOCKED pin until it goes high.
8. The devices are now locked with the control channel disabled.

24.3 Feature Availability by Device Family

All GMSL2 serializers and deserializers have a main I2C/UART control channel (Table 107).

Table 107. I2C/UART Control Channel Support by Device Family

Device Family	Main Control Channel – I2C/UART
HDMI Serializers	X
Advanced HDMI Serializers	X
DSI Serializers	X
Advanced DSI Serializers	X
CSI-2 Serializers	X
Advanced CSI-2 Serializers	X
CSI-2 Camera Deserializers	X
CSI-2 Display Deserializers	X
OLDI Deserializers	X
eDP/DP Deserializers	X
CSI-2 Quad Deserializers	X*

* CSI-2 Quad Deserializers support three independent I2C/UART ports. Each GMSL2 link provides a main control channel through which any one of the three I2C/UART ports can communicate with the serializer or peripherals connected to that link. Refer to the I2C/UART Control Channels of CSI-2 Quad Deserializers section for details.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

24.4 Pass-through Channels – I2C/UART

24.4.1 Overview

Most GMSL2 devices have two pass-through I2C/UART channels available for local or remote peripheral control. The pass-through I2C/UART channels do not have access to serializer and deserializer registers. The pass-through channels typically have the following naming convention:

- **Channel 1**:
  - SDA1_RX1
  - SCL1_TX1
- **Channel 2**:
  - SDA2_RX1
  - SCL2_TX2

Pass-through I2C and UART modes require pullup resistors (reference the Main Control Channel – I2C/UART section).

Note: The GMSL2 CSI-2 Quad Deserializers have three independent I2C/UART ports. Two of the ports are pass-through by default. These pass-through ports always provide access to local registers (a unique feature among GMSL2 devices). Refer to the CSI-2 Quad Deserializer – I2C/UART Control Channels section for details.

GMSL2 devices support touch screen applications with the pass-through I2C and GPIO features. Refer to the Touch-Screen Applications section for details.

Note: Some devices with reduced pin counts share main CC and pass-through CC functionality on the same set of pins. Extreme care must be taken when enabling and disabling the shared pins with respect to the channel being used. Refer to device-specific datasheets for pinout information. Incorrect system behavior may result if ports sharing the same pins are enabled simultaneously.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

24.4.2 Operation

The pass-through I2C/UART channels are independent of the main I2C/UART control channel. The pass-through channels provide a direct connection to remote I2C/UART peripherals but do not provide any access to internal GMSL2 device registers (except for the GMSL2 CSI-2 Quad Deserializers).

The pass-through I2C channels provide a connection to a remote I2C port without the GMSL2 devices' internal I2C register slaves hanging off the bus. This provides a mechanism to separate I2C channels and avoid multi-master conflicts.

The pass-through UART channels provide a direct point-to-point connection to the remote UART peripheral without necessitating UART Bypass Mode (as required by the main UART CC).

The pass-through channels are protected by the ARQ (Automatic Repeat Request) to improve the robustness of the channel. This is the same as the main CC. Reference the CRC Error Detection and ARQ Error Correction section for additional information.

Pass-through I2C channels support the basic Single Master I2C protocol with 7-bit slave address. Multi-master busing is not supported. The remote I2C port supports multiple slaves. The µC connected to the local I2C port must support clock stretching to accommodate possible link latency with remote peripheral access(es) (Figure 69). A time-out controller prevents locking up the I2C bus (e.g., in the case that the far side of the link fails to respond).

Pass-through UART channel data must conform to the UART Frame Format. Additionally, pass-through UART data must have a specified bit length. This requirement ensures that the remote device can create the proper UART signaling to the remote UART target peripheral device. Most devices provide two methods for specifying the bit length: manual and control channel inheritance. Note that control channel inheritance only applies if UART data has been written or read to a GMSL2 device using the properly formatted UART synchronization frame and UART accesses on the pass-through channel use the same data rate. If the UART control channel is not being used or a different data rate is desired on the pass-through UART channel(s), the pass-through channel bit length must be manually specified. GMSL2 devices support UART bit rates between 9.6Kbps and 1Mbps.

A simplified system-level block diagram for the pass-through I2C/UART feature is illustrated below (Figure 77). The local serializer/deserializer device is connected to the µC; the remote deserializer/serializer is connected to the remote peripheral I2C/UART device.

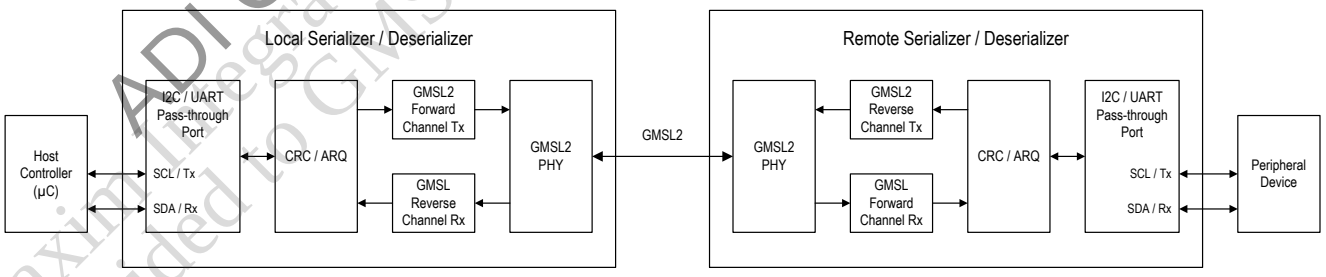


Figure 77. Pass-through I2C/UART

ADI Confidential for Review Only under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

24.4.3 Pass-through I2C

24.4.3.1 I2C Mode Configuration

The configuration of pass-through I2C channels is independent from main I2C/UART control channel configuration. The following tables contain registers associated with the pass-through I2C channels.

Note: Registers listed in the following tables may not exist in all parts. Reference device-specific datasheets and register documents for the most accurate part information.

Pass-through I2C channels are enabled on a per channel basis by setting the IIC_1_EN (Channel 1) and IIC_2_EN (Channel 2) in both the serializer and deserializer (Table 108).

Table 108. Pass-through I2C Channels Enable Registers

Bitfield	Description	Decode
IIC_1_EN[0]	Enables pass-through I2C Channel 1 (SDA1/RX1, SCL1/TX1).	0b0: I2C pass-through Channel 1 disabled 0b1: I2C pass-through Channel 1 enabled
IIC_2_EN[0]	Enables pass-through I2C Channel 2 (SDA1/RX1, SCL1/TX1)	0b0: I2C pass-through Channel 2 disabled 0b1: I2C pass-through Channel 2 enabled

The tables below contain the pass-through I2C internal slave (Table 109) and master (Table 110) configuration registers. The I2C pass-through bit rate is set with the SLV_SH_PT (local side) and MST_BT_PT (remote side) bitfields. These should be programmed according to the desired I2C pass-through bit rate to satisfy official I2C timing parameters. Set MST_DBL_PT high in the remote-side device to double the I2C-to-I2C master bit rate. The SLV_TO_PT and MST_TO_PT bits select the time-out durations which release the local- or remote-side I2C bus in case an expected response from remote side is not received within the selected time-out duration.

Table 109. Pass-Through I2C Internal Slave Configuration Registers

Bitfield	Description	Decode
SLV_SH_PT[1:0]	Pass-through I2C-to-I2C slave setup and hold time setting (setup, hold). Configures the interval between SDA and SCL transitions when driven by the internal I2C slave. Set this according to the I2C speed mode.	0b00: Set for I2C Fast-mode Plus speed 0b01: Set for I2C Fast-mode speed 0b10: Set for I2C Standard-mode speed 0b11: Reserved
SLV_TO_PT[2:0]	Pass-through I2C-to-I2C slave time-out setting. Internal GMSL2 I2C slave times out after the configured duration if it does not receive any response while waiting for a packet from the remote device.	0b000: 16us 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled

ADI Confidential for New Vision Under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential, Subject to NDA
Provided to GMSL2_Customers Under NDA

Table 110. Pass-through I2C Internal Master Configuration Registers

Bitfield	Description	Decode
MST_BT_PT[2:0]	Pass-through I2C-to-I2C master bit-rate setting. Configures the I2C bit rate used by the internal I2C master (in the device on remote side from the external I2C master).	0b000: 9.92Kbps - Set for I2C Standard- mode speed 0b001: 33.2Kbps - Set for I2C Standard-mode speed 0b010: 99.2Kbps - Set for I2C Standard- or Fast-mode speed 0b011: 123Kbps - Set for I2C Fast-mode speed 0b100: 203Kbps - Set for I2C Fast-mode speed 0b101: 397Kbps - Set for I2C Fast- or Fast-mode Plus speed 0b110: 625Kbps - Set for I2C Fast-mode Plus speed 0b111: 980Kbps - Set for I2C Fast-mode Plus speed
MST_TO_PT[2:0]	Pass-through I2C-to-I2C master timeout setting. Internal GMSL2 I2C master times out after the configured duration if it does not receive any response while waiting for a packet from the remote device.	0b000: 16us 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled
MST_DBL_PT[0]	Doubles the pass-through I2C-to-I2C master bit rate	0b0: Do not double the pass-through I2C-to-I2C master bit rate 0b1: Double the pass-through I2C-to-I2C master bit rate

Table 111 contains read-only I2C acknowledge and time-out status registers. Note that the numbers (i.e., 1 and 2) used in the bitfield names indicate whether the bit is associated with either pass-through I2C channel 1 or channel 2.

Table 111. Pass-through I2C Acknowledge and Time-Out Status (Read-Only Registers)

Bitfield	Description	Decode
REM_ACK_ACKED_1[0]	In pass-through I2C Channel 1, inverse of the I2C acknowledge bit received from remote side.	0b0: I2C acknowledge bit received as 1 0b1: I2C acknowledge bit received as 0
REM_ACK_RECVED_1[0]	In pass-through I2C Channel 1, I2C acknowledge bit for any I2C byte is received from remote side for the previous I2C packet.	0b0: I2C acknowledge bit not received 0b1: I2C acknowledge bit received
I2C_TIMED_OUT_1[0]	In pass-through I2C Channel 1, internal I2C-to-I2C slave or master has timed out while receiving packet from remote device.	0b0: Time-out has not occurred 0b1: Time-out has occurred
REM_ACK_ACKED_2[0]	In pass-through I2C Channel 2, inverse of the I2C acknowledge bit received from remote side.	0b0: I2C acknowledge bit received as 1 0b1: I2C acknowledge bit received as 0
REM_ACK_RECVED_2[0]	In pass-through I2C Channel 2, I2C acknowledge bit for any I2C byte is received from remote side for the previous I2C packet.	0b0: I2C acknowledge bit not received 0b1: I2C acknowledge bit received
I2C_TIMED_OUT_2[0]	In pass-through I2C Channel 2, internal I2C-to-I2C slave or master has timed out while receiving packet from remote device.	0b0: Time-out has not occurred 0b1: Time-out has occurred

AD Confidential for New Revision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Confidential
Provided to GMSL Customers Under NDA

The pass-through I2C/UART pin assignments can be swapped on select devices with the PT_SWAP bitfield. Reference device-specific datasheets and register documents for device support. This allows the pass-through I2C/UART pins for channel 1 (SDA1 and SCL1) to be used as pass-through I2C/UART pins for channel 2 (SDA1 and SCL2) or vice versa. Pass-through I2C/UART channels can be connected to the main CC on the remote side on a per channel basis by configuring the XOVER_EN_1 and XOVER_EN_2 bitfields.

See Table 112 for I2C/UART pin assignments and channel connections configuration registers.

Table 112. Pass-through I2C/UART Device Pin Assignments and Channel Connections Configuration Registers

Bitfield	Description	Decode
PT_SWAP[0]	Swaps I2C/UART pass-through device pin assignments	0b0: Do not swap pin assignments 0b1: Swap pin assignments
XOVER_EN_1[0]	Connects pass-through I2C/UART Channel 1 to main control channel on remote side.	0b0: Do not connect 0b1: Connect
XOVER_EN_2[0]	Connects pass-through I2C/UART Channel 2 to main control channel on remote side.	0b0: Do not connect 0b1: Connect

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

24.4.3.1.1 Pass-through I2C Address Translation

Pass-through I2C channels allow address translation for the remote-side GMSL2 device (Table 113). In the remote serializer/deserializer, the GMSL2 I2C link master recreates the I2C signaling to the remote peripheral. Additionally, with address translation, the remote I2C link master can translate a given device address into another device address with the following bitfields:

- **Channel 1**:
  - SRC_A_1 and DST_A_1
  - SRC_B_1 and DST_B_1
- **Channel 2**:
  - SRC_A_2 and DST_A_2
  - SRC_B_2 and DST_B_2

Table 113. Pass-through I2C Address Translator Configuration Registers

Bitfield	Description	Decode
SRC_A_1[6:0]	I2C address translator source A. When I2C device address matches SRC_A_1, internal I2C master replaces the device address by DST_A_1.	0bXXXXXXX: I2C address translator source A
DST_A_1[6:0]	I2C Address Translator Destination A. See the description of SRC_A_1	0bXXXXXXX: I2C address translator destination A
SRC_B_1[6:0]	I2C Address Translator Source B. When I2C device address matches SRC_B_1, internal I2C master replaces the device address by DST_B_1.	0bXXXXXXX: I2C address translator source B
DST_B_1[6:0]	I2C Address Translator Destination B. See the description of SRC_B_1	0bXXXXXXX: I2C address translator destination B
SRC_A_2[6:0]	I2C address translator source A. When I2C device address matches SRC_A_2, internal I2C master replaces the device address by DST_A_2.	0bXXXXXXX: I2C address translator source A
DST_A_2[6:0]	I2C Address Translator Destination A. See the description of SRC_A_2.	0bXXXXXXX: I2C address translator destination A
SRC_B_2[6:0]	I2C Address Translator Source B. When I2C device address matches SRC_B_2, internal I2C master replaces the device address by DST_B_2.	0bXXXXXXX: I2C address translator source B
DST_B_2[6:0]	I2C Address Translator Destination B. See the description of SRC_B_2.	0bXXXXXXX: I2C address translator destination B

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Available Only to Maxim Integrated Company Confidential, Under NDA Provided to GMSL2_Customers

24.4.4 Pass-through UART

24.4.4.1 UART Mode Configuration

The following tables contain bitfields associated with the pass-through UART channels. Note that the PT_SWAP, XOVER_EN_1, and XOVER_EN_2 bitfields (only available on select devices) are common to the pass-through I2C and UART channels. See Table 112 above for configuration information.

Note: Bitfields listed in the following tables may not exist in all parts. Reference device-specific datasheets and register documents for the most accurate part information.

Pass-through UART channels are enabled on a per channel basis by setting the UART_1_EN (Channel 1) and UART_2_EN (Channel 2) in both the serializer and deserializer (Table 114).

Table 114. Pass-Through UART Channels Enable Registers

Bitfield	Description	Decode
UART_1_EN[0]	Enables pass-through UART Channel 1 (SDA1/RX1, SCL1/TX1).	0b0: Pass-through UART Channel 1 disabled 0b1: Pass-through UART Channel 1 enabled
UART_2_EN[0]	Enables pass-through UART Channel 2 (SDA2/RX2, SCL2/TX2).	0b0: Pass-through UART Channel 2 disabled 0b1: Pass-through UART Channel 2 enabled

The UART parity bit can be enabled or disabled in pass-through operation on a per channel basis (Table 115).

Table 115. Pass-Through UART Parity Check Configuration Registers

Bitfield	Description	Decode
DIS_PAR_1[0]	Disables parity bit in pass-through UART (Channel 1)	0b0: Parity bit enabled 0b1: Parity bit disabled
DIS_PAR_2[0]	Disables parity bit in pass-through UART (Channel 2)	0b0: Parity bit enabled 0b1: Parity bit disabled

If the main UART CC is being used and UART data has been written or read to a GMSL2 device (serializer or deserializer) using the properly formatted UART synchronization frame in base mode, pass-through UART channels use the same data rate automatically detected by the GMSL2 device in base mode.

If the main UART CC is not being used, or if a different data rate is required on the pass-through UART channels, the pass-through UART channels bit length must be manually programmed. Bit length is manually configured by first setting BITLEN_MAN_CFG_1/2 for the appropriate channel(s). Then, the 14-bit bitlength is specified in terms of the internal 150MHz clock with the concatenation of bitfields BITLEN_PT_1_H[5:0] / BITLEN_PT_1_L[7:0] for channel 1 and BITLEN_PT_2_H[5:0] / BITLEN_PT_2_L[7:0] for channel 2 (Table 116).

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

Table 116. Pass-Through UART Bit Rate Configuration Registers

Bitfield	Description	Decode
BITLEN_MAN_CFG_1[0]	Uses the custom UART bit rate (selected by the BITLEN_PT_1_L and BITLEN_PT_1_H bitfields) in pass-through UART Channel 1.	0b0: Use standard bit rate 0b1: Use custom bit rate
BITLEN_PT_1_L[7:0]	Low byte of custom UART bit length for pass-through UART Channel 1. Set this register to the UART bit length divided by 6.666ns (LSB 8 bits). Set BITLEN_MAN_CFG_1 to 1 to use this value.	0xXXXXXXXX: Low byte of custom UART bit length for pass-through UART Channel 1
BITLEN_PT_1_H[5:0]	High byte of custom UART bit length for pass-through UART Channel 1. Set this register to the UART bit length divided by 6.666ns (MSB 6 bits). Set BITLEN_MAN_CFG_1 to 1 to use this value.	0xXXXXXX: High byte of custom UART bit length for pass-through UART Channel 1
BITLEN_MAN_CFG_2[0]	Uses the custom UART bit rate (selected by the BITLEN_PT_2_L and BITLEN_PT_2_H bitfields) in pass-through UART Channel 2.	0b0: Use standard bit rate 0b1: Use custom bit rate
BITLEN_PT_2_L[0]	Low byte of custom UART bit length for pass-through UART Channel 2. Set this register to the UART bit length divided by 6.666ns (LSB 8 bits). Set BITLEN_MAN_CFG_2 to 1 to use this value.	0xXXXXXXXX: Low byte of custom UART bit length for pass-through UART Channel 2
BITLEN_PT_2_H[0]	High byte of custom UART bit length for pass-through UART Channel 2. Set this register to the UART bit length divided by 6.666ns (MSB 6 bits). Set BITLEN_MAN_CFG_2 to 1 to use this value.	0xXXXXXX: High byte of custom UART bit length for pass-through UART Channel 2

Overflow conditions are monitored for each UART pass-through channel. UART_RX_OVERFLOW_1/2 and UART_TX_OVERFLOW_1/2 are read-only bitfields that latch if an overflow condition has occurred with UART pass-through communications across the serial link (Table 117).

Note: In rare circumstances, a link reset during a UART communication or insufficient serial link bandwidth may result in overflow conditions.

Table 117. Pass-Through UART Rx/Tx FIFO Overflow Status (Read-Only Registers)

Bitfield	Description	Decode
UART_RX_OVERFLOW_1[0]	Pass-through UART RX FIFO overflow. Set to 1 following an overflow condition. Clears upon read.	0b0: No overflow occurred 0b1: Overflow occurred
UART_TX_OVERFLOW_1[0]	Pass-through UART TX FIFO overflow. Set to 1 following an overflow condition. Clears upon read.	0b0: No overflow occurred 0b1: Overflow occurred
UART_RX_OVERFLOW_2[0]	Pass-through UART RX FIFO overflow. Set to 1 following an overflow condition. Clears upon read.	0b0: No overflow occurred 0b1: Overflow occurred
UART_TX_OVERFLOW_2[0]	Pass-through UART TX FIFO overflow. Set to 1 following an overflow condition. Clears upon read.	0b0: No overflow occurred 0b1: Overflow occurred

ADI Confidential for New Vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

24.4.5 Pass-through I2C/UART in Splitter Mode

The pass-through I2C and UART channels operate point-to-point. This differs from the bus operation of the main CC. For systems with a GMSL2 link in Splitter Mode or Reverse Splitter Mode, a separate pass-through I2C/UART channel is required for each link to provide a connection to remote I2C/UART peripherals. For example, two deserializers, each with connected peripherals, are connected to a serializer in splitter mode. If pass-through I2C/UART Channel 1 is enabled on the deserializer connected to Link A, pass-through I2C/UART Channel 2 must be enabled on the deserializer connected to Link B.

24.4.6 Pass-through Channels Debug Techniques

The pass-through I2C/UART channels can be inspected with a logic analyzer or digital oscilloscope. If there are issues with the pass-through channels, verify that the waveforms observed on the SDA1_RX1 and SCL1_TX1 pins (channel 1) and SDA2_RX1 and SCL2_TX2 pins (channel 2) are as expected. Reference the I2C timing diagram in the device datasheet for additional information.

Ensure that the pass-through I2C/UART channel is enabled in both the serializer and deserializer. For example, for pass-through I2C channel 1, the IIC_1_EN register must be set to 1 in both the serializer and deserializer.

For pass-through I2C channels, there are several methods to verify that I2C Ack bits are received from the remote side for any transmitted I2C byte. The Ack bit after each I2C byte can be observed with a logic analyzer or digital oscilloscope. Alternatively, the REM_ACK_RECVED_1 and REM_ACK_RECVED_2 bitfields read 1 if an I2C Ack bit has been received from the remote side for the previous I2C byte, and REM_ACK_ACKED_1 and REM_ACK_ACKED_2 can be read back to see the value of the received Ack bit.

The I2C_TIMED_OUT_1 and I2C_TIMED_OUT_2 registers can be read back to check if the internal I2C-I2C master or slave has timed-out while receiving packets from the remote device on pass-through I2C channel 1 or 2.

UART overflow conditions are flagged independently for each channel with the following bitfields: UART_RX_OVERFLOW_1 / UART_RX_OVERFLOW_2 and UART_TX_OVERFLOW_1 / UART_TX_OVERFLOW_2.

ADI Confidential for New Version under NDA
Analog Devices Confidential Information only for
Maxim Integrated Company Confidential,
Provided to GMSL2_Customer Under NDA

24.5 Feature Availability by Device Family

All GMSL2 serializers and deserializers have two pass-through I2C/UART channels (Table 118).

Table 118. Pass-through I2C/UART Channel Support by Device Family

Device Family	Pass-Through I2C/UART Channels
HDMI Serializers	X
Advanced HDMI Serializers	X
DSI Serializers	X
Advanced DSI Serializers	X
CSI-2 Serializers	X
Advanced CSI-2 Serializers	X
CSI-2 Camera Deserializers	X
CSI-2 Display Deserializers	X
OLDI Deserializers	X
eDP/DP Deserializers	X
CSI-2 Quad Deserializers	X*

* Each link of CSI-2 Quad Deserializers supports a dedicated control channel and two additional tunneled pass-through I2C/UART channels. One of the pass-through channels can be configured as the main control channel by control channel crossover feature. Refer to the I2C/UART Control Channels of CSI-2 Quad Deserializers section for details.

ADI Confidential for New Version under NDA
Analog Devices Confidential, Limited Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

25 Touch-Screen Applications

25.1 Overview

GMSL2 devices can support touch-screen applications with GPIO forwarding and I2C. Reference the I2C/UART and General-Purpose Input and Output (GPIO) sections for detailed information on these features.

Note: It is recommended to use the pass-through I2C channel for touch communication to avoid potential I2C collisions or delays due to register configuration and error handling on the main I2C control channel bus.

A typical touch-screen system comprising one controller and a connected touch screen is shown below (Figure 78).

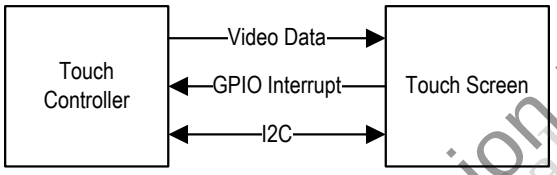


Figure 78.Typical Touch-Screen System

Touch-screen systems can be scaled by driving multiple touch screens from a single controller. The controller shown below has dedicated I2C and GPIO channels for each touch screen (Figure 79).

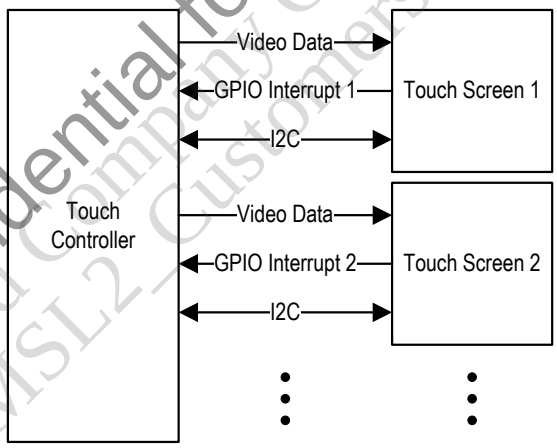


Figure 79. Touch-Screen System with Multiple Touch-Screen Interfaces

Note: To reduce device pin utilization in applications with multiple touch screens, the GPIO interrupt channel(s) and I2C channel(s) can be shared or isolated through an OR logic gate. Reference the *Multiple Touch-Screens Application Example* for implementation details.

ADI Confidential for New Vision under NDA
Analog Devices Initial Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

25.2 Configuration

25.2.1 Initial Setup

It is recommended to independently validate the touch-screen system without the GMSL I2C or GPIO channels by connecting the GPIO interrupt and I2C directly between the touch screen and touch controller (*Figure 80*). This simplified architecture isolates the touch-screen system from any issues related to GMSL and allows for a more accurate and efficient debugging process.

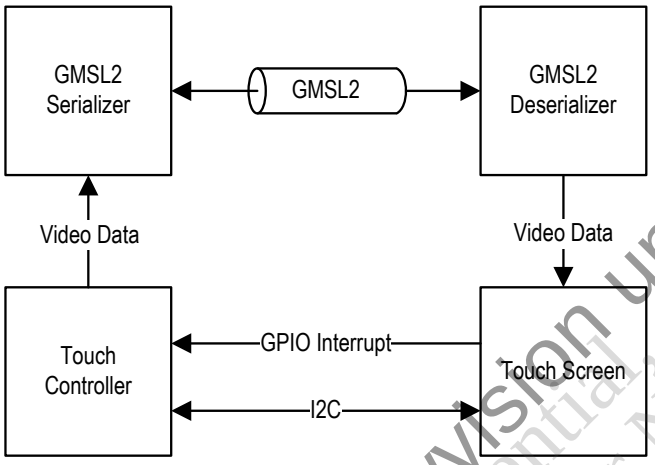


Figure 80. Recommended Setup for Initial Touch-Screen System Verification

After initial validation, the touch-screen system can be integrated into a GMSL2 system with GMSL pass-through I2C channels and GPIO forwarding (Figure 81). Note that GMSL2 system requirements and the potential impacts on other system features must be evaluated while allocating resources (e.g., GPIO pins and I2C channels) to the touch-screen data buses. Important considerations include how many touch-screen interfaces will be implemented in the system and how the system will handle the GPIO interrupts and I2C communications. Reference the Touch-Screen GPIO Forwarding and Touch-Screen I2C sections for more information.

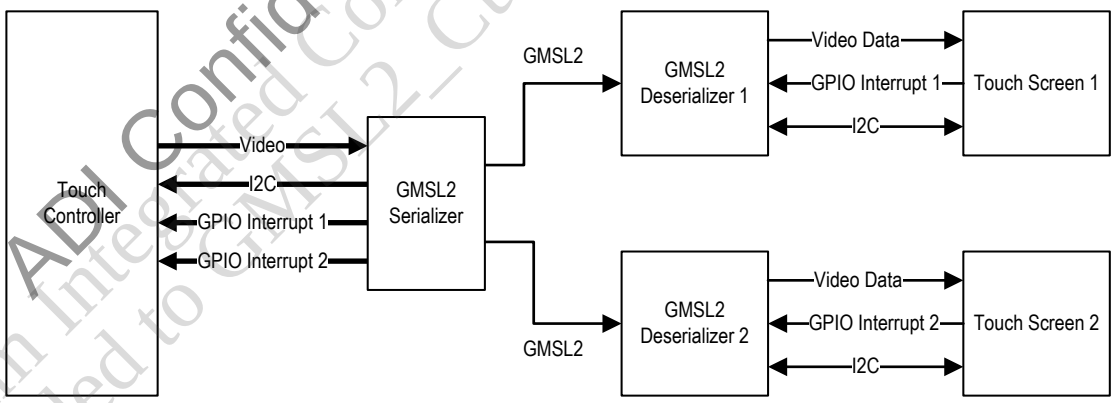


Figure 81. Typical GMSL2 Touch-Screen Application

ADI Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
Newversion under NDA

25.2.2 Touch-Screen GPIO Forwarding

The information below is intended to provide instructions relevant to touch-screen applications. Refer to the Configuring GPIO Forwarding section for detailed configuration instructions and additional guidelines.

There are two implementations of GPIO forwarding in touch-screen applications:

- **Combined** – GPIO pins are wired together and routed through an OR gate. The system is notified of a touch interaction if any of the driven touch screens are touched. The I2C data buses for all connected touch screens are read to determine the origin of the touch interaction. Reference Figure 85.
- **Isolated** – All GPIO communications are isolated and routed to separate pins. The I2C channel reads the touch-screen display corresponding to the GPIO communicating a touch interaction. Reference Figure 81.

Note: GPIO pin mappings vary by device and pins may have shared functionality. Ensure that GPIO pin(s) used in touch-screen applications do not overlap with any other functions or features used in the system. Refer to device-specific datasheets for individual device support.

Configure registers GPIO_A, GPIO_B, and GPIO_C for each GPIO channel used in both the serializer and deserializer device.

The GPIO_A register is used to declare if the GPIO channel will be used as an input or output GPIO. For touch-screen applications in GMSL systems, the GPIO on the deserializer is an input that transmits to the serializer, and the GPIO on the serializer is an output that receives from the deserializer. If a GPIO pin is configured as an input, the output of the GPIO must be disabled by setting the GPIO_OUT_DIS[0] bit to 1. The GPIO is set up for transmission by programming GPIO_TX_EN[0] to 1 and GPIO_RX_EN[0] to 0. If a GPIO pin is configured as an output, the output of the GPIO must be enabled by setting the GPIO_OUT_DIS[0] bit to 0. The GPIO is set up for reception by setting GPIO_TX_EN[0] to 0 and GPIO_RX_EN[0] to 1.

The GPIO_B register assigns the GPIO transmission ID used for mapping GPIO pins across the serial link. The default transmission ID corresponds with the GPIO channel number. The OUT_TYPE[5] bit programs the driver type as either open-drain or push-pull. The PULL_UPDN_SEL[7:6] bits set the buffer pull-up/pull-down configuration.

The GPIO_C register assigns the GPIO pin ID while receiving. This programs the ID the GPIO will look for to recognize the appropriate GPIO signals from the transmitting device. The default reception ID corresponds with the GPIO channel number.

Note: The default transmission and reception ID settings connect and sync GPIOs with the same channel number across the serial link. Any GPIO ID reassignment must be coordinated between the transmitter and receiver devices to ensure proper operation of GPIO communications.

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers under NDA

25.2.3 Touch-Screen I2C

The information below is intended to provide instructions relevant to touch-screen applications. Refer to the Main I2C Control Channel and Pass-through I2C sections for detailed configuration instructions and additional guidelines.

GMSL2 devices have three available I2C channels, including the main I2C control channel used for internal device register access and two pass-through channels. Final design requirements and system features must be considered when deciding which I2C channel(s) to use for touch-screen applications. It is recommended that touch screens be given a dedicated I2C pass-through channel to eliminate possible latency resulting from other features using the channel.

Note: I2C is used for internal register access and control of GMSL2 devices as well as communication with peripherals. Reference the I2C/UART section for more information.

The I2C pass-through channels are individually enabled by setting IIC_1_EN[0] (channel 1) or IIC_2_EN[0] (channel 2) in both the serializer and deserializer devices.

Note: I2C pin mappings and the MFP assignment for the I2C pass-through channels vary by device. Refer to device-specific datasheets for individual device support.

ADI Confidential for Newvision under NDA
Analog Devices Confidential - Distribution Only to Maxim Integrated Company Confidential Provided to GMSL2_Customers Under NDA

25.3 Single Touch-Screen Application Example

Figure 82 shows a touch-screen system consisting of a single touch screen driven by a controller (raspberry pi) via one I2C pass-through channel and one forwarded GPIO through a GMSL link. In this example, GPIO2 is designated as the GPIO, and I2C pass-through channel 2 is configured to be the I2C pass-through channel. The system is programmed to read the I2C data bus for the touch screen after receiving a GPIO interrupt.

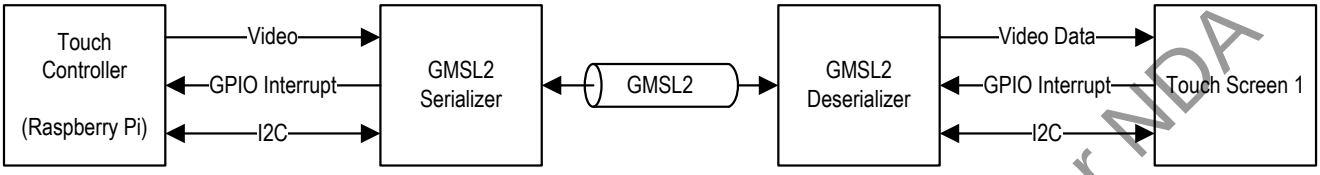


Figure 82. Single Touch Screen Setup

25.3.1 Single Touch Screen Example Script

The following script applies to a GMSL2 system comprising an OLDI deserializer and an HDMI serializer.

```
########################## TOUCH SCREEN APPLICATION ##########################
# GPIO2 was selected to not interfere with the functions on GPIO1.           #
# I2C passthrough 2 was chosen to allow a dedicated I2C channel              #
# while not interfering with audio communication.                            #
##############################_SER_MAX96751@0x80_##############################
#enable I2C PASSTHRU 2           Enable I2C passthrough channel 2
0x80,0x01,0xE8
#GPIO2 set up as receive         GPIO2 passthrough set up to be receiver
0x80,0x206,0x84
#GPIO2 B set up                  GPIO2 out type is push-pull with pull down resistor
0x80,0x207,0xA2
#GPIO2 C set up                  GPIO2 RX ID 2
0x80,0x208,0x02
##############################_DES1_MAX96752@0x90_##############################
#disable AUD_TX_EN               Disable Audio TX to free up I2C PT pin
#0x90,0x02,0x43
#enable I2C PASSTHRU 2           Enable I2C PT 2
#0x90,0x01,0x22
#enable OLDI                     Enable and setup OLDI display settings
0x90,0x1CE,0x5E
#GPIO2 set up as transmit        GPIO2 set up as transmitter
0x90,0x206,0x83
#GPIO2 B set up                  GPIO2 TX ID 2
0x90,0x207,0xA2
#GPIO2 C set up                  GPIO2 RX ID 2
0x90,0x208,0x42
```

Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA
Analog Devices Confidential Distribution only
NDA Confidential for review under NDA

25.3.2 Single Touch Screen Example Analysis

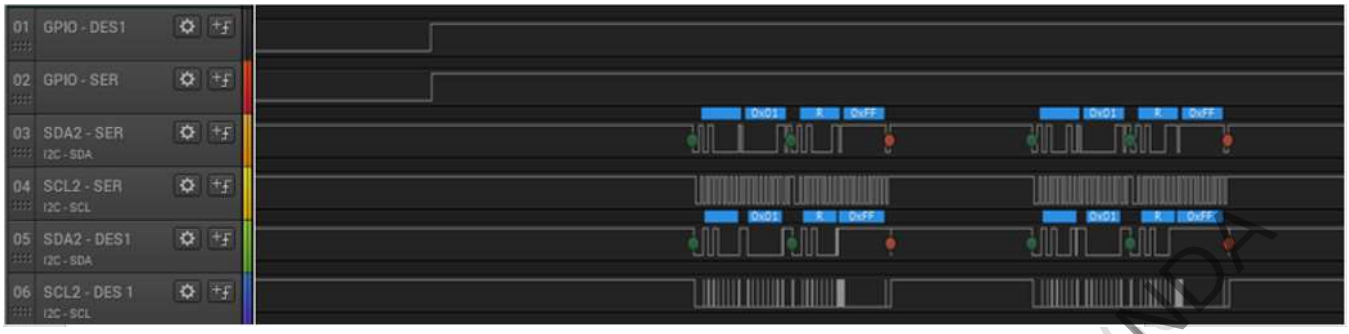


Figure 83. Single Touch Screen Example Signal Analysis

The signal analysis of the single touch screen example is presented in Figure 83. The GPIO signal shown in the top trace (GPIO – DES1) is the interrupt input to the deserializer from the touch screen. When this is active, the corresponding GPIO on the serializer (GPIO – SER) passes the signal through to the main system. After receiving the GPIO interrupt, the system reads the I2C data from the touch screen via the pass-through I2C channel.

ADI Confidential for Newvision Under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

25.3.3 Single Touch Screen Timing Analysis

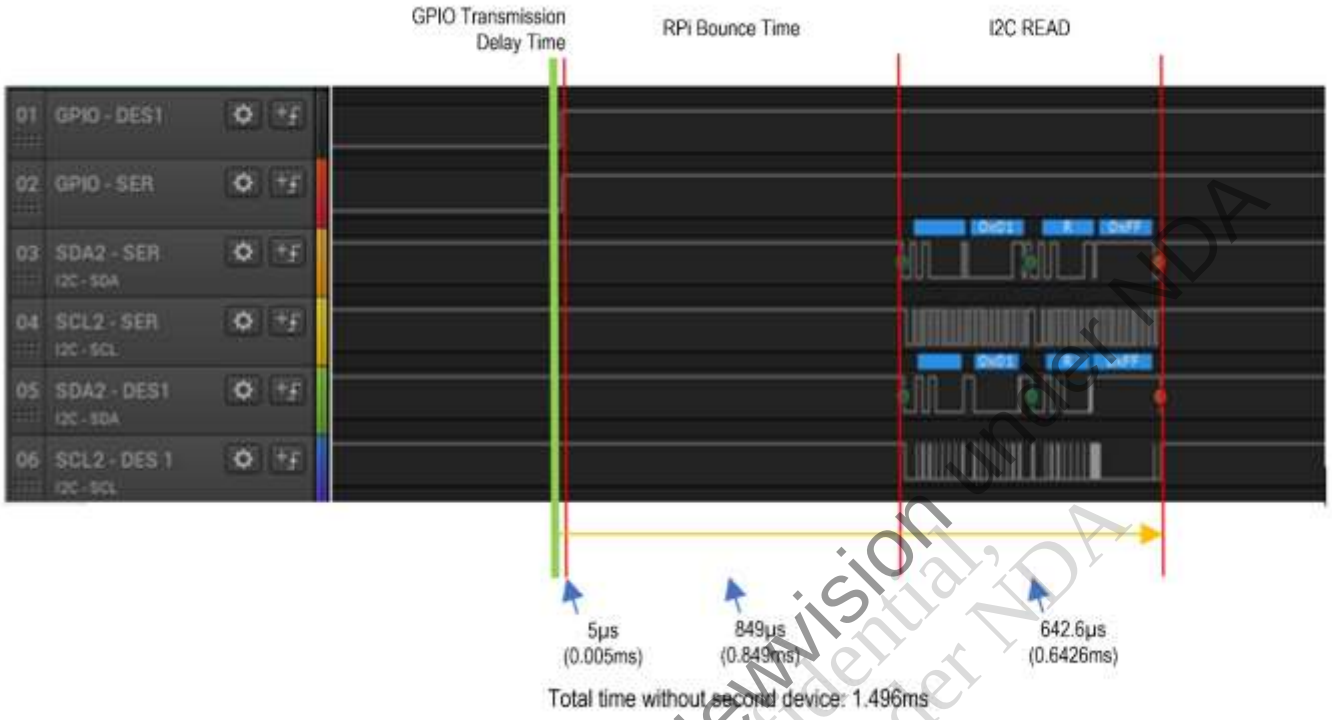


Figure 84. Single Touch Screen Example Timing Analysis

Figure 84 shows the timing analysis of the single touch screen example. The forwarded GPIO transmission incurs a negligible 5µs transmission delay that does not result in noticeable latency. The controller (raspberry pi) uses the 0.849ms bounce time to prevent any unwanted noise or jitter from triggering the I2C read more than once within a 1ms interval. This improves accuracy and mitigates unnecessary I2C reads that may overcrowd the I2C data bus. The subsequent I2C read for the device takes approximately 0.6ms. The combined time required to communicate with one touch screen is 1.5ms. This 1.5ms data refresh and sample period means that there will be no noticeable latency or delay when interacting with the device.

ADI Confidential for Review under NDA
Analog Devices Confidential, Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA

25.4 Multiple Touch-Screens Application Example

Figure 85 shows a touch-screen application consisting of two touch screens being driven by a controller (raspberry pi) via two I2C pass-through channels and two forwarded GPIOs through a GMSL link. Each touch screen is allocated dedicated I2C channel and a forwarded GPIO. This example uses an OR gate to feed both GPIO pins from the GMSL serializer onto one system GPIO pin. Note that GMSL system GPIO signals are edge-triggered; the GPIO will not behave as an OR gate if two GPIO Tx pins to be received by the Rx GPIO have the same Tx ID. The system is programmed to read the I2C data bus on both touch screens after receiving any GPIO interrupt.

Optimal system designs will allocate a dedicated GPIO pin to each touch-screen interface. The configuration used in this example reduces the number of required GPIO pins from the system with the OR gate and simulates a heavy use-case application. In applications where both touch screens are actively used, both I2C data buses will need to be read frequently. Feeding both GPIO pins into an OR gate causes the I2C data buses to be read as if both touch screens were being used simultaneously.

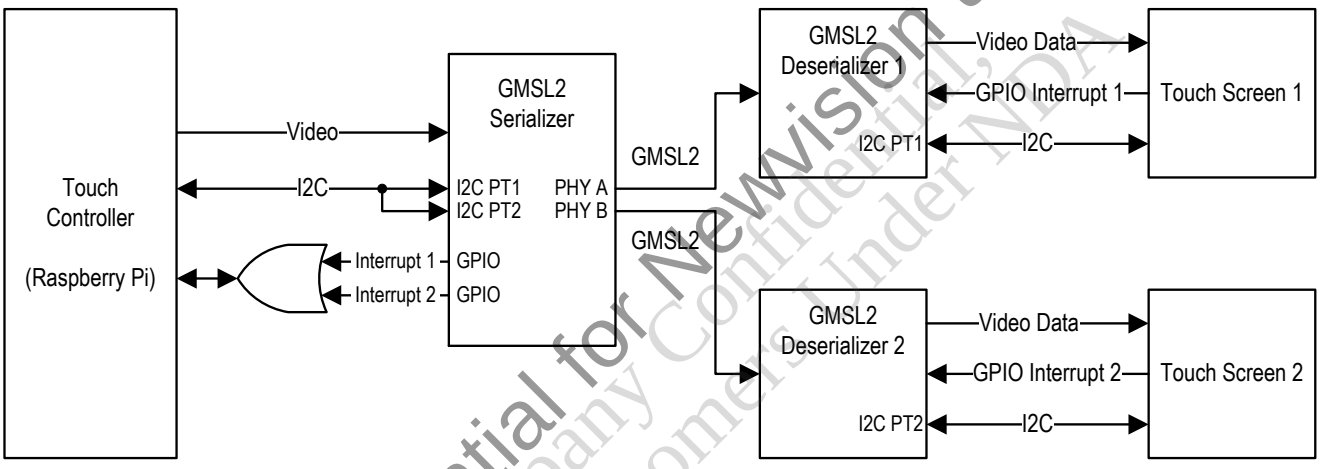


Figure 85. Multiple Touch Screen Setup

ADI Confidential for Newvision under NDA
Analog Devices Confidential, only
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2_Customers Under NDA

25.4.1 Multiple Touch Screens Example Script

```
######################### TOUCH SCREEN APPLICATION ##########################
############################_SER_MAX96751@0x80_################################
#set splitter mode, auto link, and one shot reset (Sets up Splitter Mode and Resets Device)
0x80,0x10,0x23
delay 100
#Splitter mode video set up with video replication for example purposes
#VIDEO X TX_SPLT_MASK_A Packets are transmitted of GMSLA in splitter
0x80,0x53,0x30
#HDMI AUTOS
0x80,0x01,0xC8

#enable I2C PASSTHRU 1 and PASSTHRU 2   Enable I2C passthrough channel 1 and 2
0x80,0x01,0xF8
#I2C PT TX SPLT_MASK_A[0] = 1     Enable I2C PT1 to only go through PHY A
0x80,0xA3,0x10
#I2C PT TX SPLT_MASK_B[0] = 1     Enable I2C PT2 to only go through PHY B
0x80,0xAB,0x20
#GPIO6 set up as receive          GPIO6 passthrough set up to be receiver
0x80,0x212,0x84
#GPIO6 B set up                   GPIO6 out type is push-pull with pull down resistor
0x80,0x213,0xA6
#GPIO6 C set up                   GPIO6 RX ID 1
0x80,0x214,0x06
#GPIO7 set up as receive          GPIO7 passthrough set up to be receiver
0x80,0x215,0x84
#GPIO7 B set up                   GPIO7 out type is push-pull with pull down resistor
0x80,0x216,0xA7
#GPIO7 C set up                   GPIO7 RX ID 2
0x80,0x217,0x07

############################_DES1_MAX96752@0x90_################################
#enable I2C PASSTHRU 1            Enable I2C PT 1
0x90,0x01,0x12
#GPIO6 set up as transmit         GPIO6 set up as transmitter
0x90,0x212,0x83
#GPIO6 B set up                   GPIO6 TX ID 2
0x90,0x213,0xA6
#GPIO6 C set up                   GPIO6 RX ID 2
0x90,0x214,0x46
#enable OLDI                      Enable and setup OLDI display settings
0x90,0x1CE,0x5E
############################_DES2_MAX96752@0x94_################################
#disable AUD_TX_EN                Disable Audio TX to free up I2C PT pin
0x94,0x02,0x43
#enable I2C PASSTHRU 2            Enable I2C PT 2
0x94,0x01,0x22
#GPIO7 set up as transmitter      GPIO7 set up as transmitter
0x94,0x215,0x83
#GPIO7 B set up                   GPIO7 TX ID 2
0x94,0x216,0xA7
#GPIO7 C set up                   GPIO7 RX ID 2
0x94,0x217,0x47
#enable OLDI                      Enable and setup OLDI display settings
0x94,0x1CE,0x5E
```

ADI Confidential for New vision under NDA
Maxim Restricted Distribution pursuant to NDA
Provided to GMSL2 Integrated Company Confidential, Customers Under NDA

25.4.2 Multiple Touch Screens Example Analysis



Figure 86. Multiple Touch Screen Example Signal Analysis

The signal analysis of the multiple touch screen example is shown in Figure 86. In this example, interaction with either touch screen triggers a GPIO interrupt to the connected deserializer (represented by GPIO – DES1 and GPIO – DES2). The corresponding GPIO on the serializer passes the signal though the OR gate to the system. The system takes the GPIO interrupt and reads the I2C data from both touch screens via the pass-through I2C channels on both deserializers.

This example has many similarities to the Single Touch-Screen Application Example. There are two independent GPIO channels that connect to each touch screen. The GPIO interrupts pass through an OR gate before being received on a single GPIO pin on the system. The system does not determine the source of the interrupt and reads the I2C data of both touch screens. The touch screens are assigned separate pass-through I2C channels. So, the system reads the data buses of both I2C channels whenever a GPIO interrupt is received.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribution only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

25.4.3 Multiple Touch Screens Timing Analysis



Figure 87. Multiple Touch Screen Example Timing Analysis

Figure 87 shows the timing analysis of the multiple touch screen example. The forwarded GPIO transmission incurs a negligible 5µs transmission delay that does not result in noticeable latency. The controller (raspberry pi) uses the 0.849ms bounce time to prevent any unwanted noise or jitter from triggering the I2C read more than once within a 1ms interval. This improves accuracy and mitigates unnecessary I2C reads that may overcrowd the I2C data bus. Then, the I2C read for the first device takes approximately 0.6ms. The combined time required to communicate with one touch screen is 1.5ms. This 1.5ms data refresh and sample period means that there will be no noticeable latency or delay when interacting with the device.

This system has two connected touch screens with independent pass-through I2C channels; however, only one I2C pin is used on the system side. After completing the I2C read for the first touch screen, the system pauses for 0.5ms. Then, the I2C read for the second device starts and takes approximately 0.6ms. The total time to communicate with both touch screens is 2.5ms. Note that there will be a minimum of 1ms delay added for each additional touch screen added to the same I2C bus.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Proprietary
Maxim Integrated Confidential, Proprietary
Provided to GMSL2_Customers Under NDA

26 I/O Matrix (MFP)

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

27 General-Purpose Input and Output (GPIO)

27.1 Overview

All GMSL2 devices have multi-function pins (MFP) that can be used as general-purpose input and output (GPIO) pins or for other functionality (e.g., I2C, I2S, SPI, etc.). This section explains the GPIO function of MFP pins. Note that the I/O Matrix (MFP) is device-specific; refer to device datasheets for individual device support.

The GPIO blocks of GMSL2 devices communicate and regenerate state changes of GPIO pins from one side of the serial link to the other. GMSL2 serializers and deserializers each contains one transmitter block (GPIO Tx) and one receiver block (GPIO Rx). The transmit block collects the changes from GPIO pins and transmits them to other side of the serial link; the receive block at the other side of the link regenerates the observed changes at the corresponding pins.

27.2 Operation

GPIO pin mapping is coordinated across the serial link via GPIO "pin ID" assignments. Each GPIO input is assigned a pin ID that is included in the packet sent across the serial link and corresponds with a GPIO output. The pin ID of each GPIO transition packet sent from the input is compared with the mapping on the receive-side device, and the transition is reflected on the corresponding output GPIO pin. By default, the GPIO mapping is GPIO0–GPIO0, GPIO1–GPIO1, GPIO2–GPIO2, etc. The GPIO mappings can be changed via registers.

GMSL2 devices use 5-bit pin IDs that can support mapping up to 32 GPIO pins. Note that the usable number of GPIOs is limited by the device specific GPIO pinout. GPIOs are matched with the pins in the I/O matrix for each part number. Refer to datasheets for individual device GPIO support.

Delay compensation can be configured for GPIO packet transmissions. With delay compensation enabled, GPIO packets are transmitted over the serial link and a configurable fixed latency is applied before the changes are observed on the receive-side GPIO pins. This fixed latency mitigates variable delay that is observed as jitter. In standard mode (without delay compensation), GPIO packets are sent as soon as possible based on available link bandwidth, and changes observed as they are received by the device on the other side of the link. Standard mode offers lower latency than delay compensated mode, however, transmissions may incur variable delay (jitter).

Additionally the GPIO transmit priority over the link can be adjusted via the priority-based scheduler settings.

27.2.1 Initialization

1. **On link lock**: The GPIO Tx block sends the initial values of enabled GPIO Tx pins upon the establishment of link lock regardless of the pin states.
2. **In sleep mode and on wake up from sleep mode**: The GPIO Rx block drives enabled GPIO Rx pin(s) with the last value received before it entered sleep mode.

27.2.2 GPIO Transmit Control

The GPIO transmit control block is responsible for collecting, scheduling, and transmitting the requests from GPIO pins. The requests are scheduled in the order in which they are received; the queue operates on a first-come-first-served basis.

Analog Devices Confidential for Review Purposes only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

27.2.3 GPIO Delay Compensation

GPIO delay compensation mode is used to minimize the jitter of the transmitted signal by adding a fixed delay to every GPIO transition on the receive side. This is particularly useful for pulse generation where timing is critical.

Delay compensation allows input transitions to be observed at the output with a fixed latency (with up to ±10ns variation). This latency is configurable via registers.; however, the default latency value should be used for typical use cases.

In delay compensation mode, a buffer is used to store the transitions before they are reflected on the output pin. This buffer can store up to 15 transitions. The maximum delay compensation value is limited by this buffer size. For instance, if the data rate of the input signal is 4Mbps, the maximum delay compensation value is 3.5µs.

27.2.4 Bidirectional Operation

GMSL2 devices support bidirectional GPIO operation. This capability allows control channel and event-driven activity of peripheral devices on each side of the link to be transmitted across the serial link via the GPIO pins. For bidirectional operation, the GPIO Rx block disables the GPIO Tx block for a short duration in order to prevent false transmissions while the received value is driven to the GPIO pin.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution for Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

27.3 Configuration

Each GPIO is controlled by three registers: GPIO_A, GPIO_B, and GPIO_C. In the register documentation, the GPIO mapping is sequential (i.e., the first three GPIO registers correspond to GPIO0, then next three to GPIO1, etc.) The GPIOs are mapped to different MFP pins in the package depending on the part number.

When programming GPIOs, it is important to program the GPIO Rx before the GPIO Tx to avoid asynchronous initial states. For example, if Tx is low but Rx is high, the first transition of Tx from low to high will be ignored by Rx since Rx is already high. All subsequent transitions will be correctly observed.

27.3.1 GPIO Pull-Up and Pull-Down Resistor Setup

Each GPIO can be programmed to have either a pull-up, pull-down, or no resistor. The pull-up or pull-down resistance can be set to either 40kΩ or 1MΩ.

The resistor is configurated with the PULL_UPDN_SEL[1:0] register:

- 00: No resistor
- 01: Pull-up resistor
- 10: Pull-down resistor
- 11: Reserved

The resistance value of the resistor is set using the RES_CFG register:

- 0: 40kΩ
- 1: 1MΩ

Note: Register RES_CFG is ignored if the resistor is disabled (PULL_UPDN_SEL[1:0] = 00).

27.3.2 GPIO Output Driver Setup

The GPIO output driver can be enabled or disabled. When enabled, the output driver can be configured to be either open drain or push-pull. The output driver is enabled by writing GPIO_OUT_DIS = 0 and disabled by writing GPIO_OUT_DIS = 1. The output driver is configured for open drain mode (i.e., NMOS output driver enabled) by writing OUT_TYPE = 0 and for push-pull mode (i.e., both NMOS and PMOS output driver enabled) by writing OUT_TYPE = 1.

27.3.3 Configuring GPIO Forwarding

GPIO forwarding is the transmission and regeneration of state changes of GPIO pins on the local side of the serial link to the corresponding GPIO pins on the remote side. To ensure reliable forwarding, the local and remote side GPIOs must be properly configured. Each GPIO has a register-configurable GPIO_TX_ID and GPIO_RX_ID that are used for mapping GPIO pins across the serial link. Note that this configuration applies to both serializer-to-deserializer and deserializer-to-serializer communications.

Analog Confidential for New vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

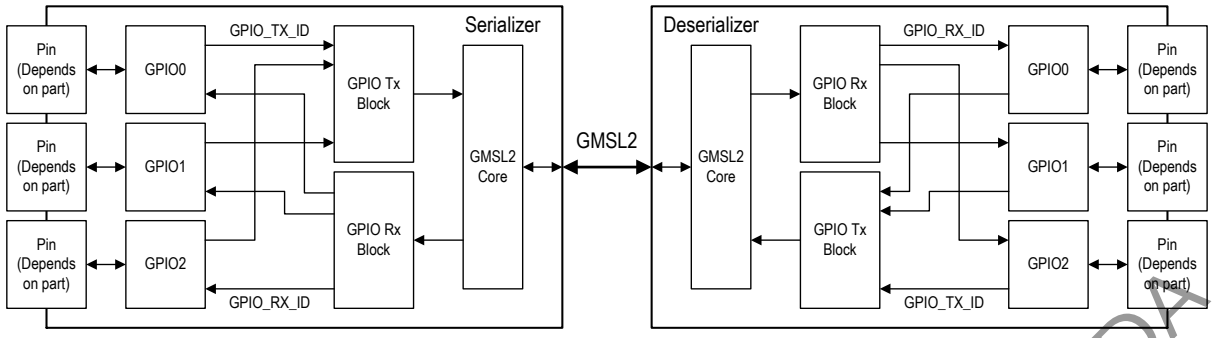


Figure 88. Block Diagram of GPIO Tx and Rx Blocks Mapping

27.3.3.1 Individual GPIO Pin Configuration

Configuring Input GPIO:

- Set GPIO_TX_ID with a value from 0–31 to assign the GPIO pin ID.
- Write GPIO_TX_EN = 1 to enable the GPIO transmit block.

Configuring Output GPIO:

1. Set GPIO_RX_ID with a value from 0–31 to assign the GPIO pin ID. This must be the same value used for GPIO_TX_ID in order to map the input and output GPIO pins.
2. Write GPIO_RX_EN = 1 to enable the GPIO receive block for the GPIO pin.

By default, the GPIO_TX_ID and GPIO_RX_ID are the same value as the GPIO number. For example, the default GPIO_TX_ID and GPIO_RX_ID values for GPIO1 is 1; accordingly, GPIO1 is mapped to GPIO1 on the opposite side of the serial link by default. Transitions on a single GPIO input can also be mapped to multiple GPIO outputs (broadcasting). This is configured by setting multiple GPIOs to the same GPIO_RX_ID.

27.3.4 Delay Compensated Mode

Delay compensation mode is used to minimize the jitter of the transmitted signal by adding a fixed delay to every GPIO transition. Delay compensated mode is enabled by setting TX_COMP_EN = 1 on the input GPIO. GPIO_FWD_ CDLY[5:0] is used to configure the forward compensation delay and GPIO_REV_ CDLY[5:0] is used to configure the reverse compensation delay. Typically, the default forward- and reverse-channel compensation delay values are recommended and should only be increased if required by the application.

The total forward channel compensation delay has a default value of 3.4µs and is calculated with the following equation:

$$(GPIO_FWD_CDLY[5:0] + 1) * 1.7\mu s$$

The total reverse channel compensation delay has a default value 15.3µs and is calculated with the following equation:

$$(GPIO_REV_CDLY[5:0] + 1) * 1.7\mu s$$

ADI Confidential for New Revision under NDA
Analog Devices Inc. Distribute only under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers_Under_NDA

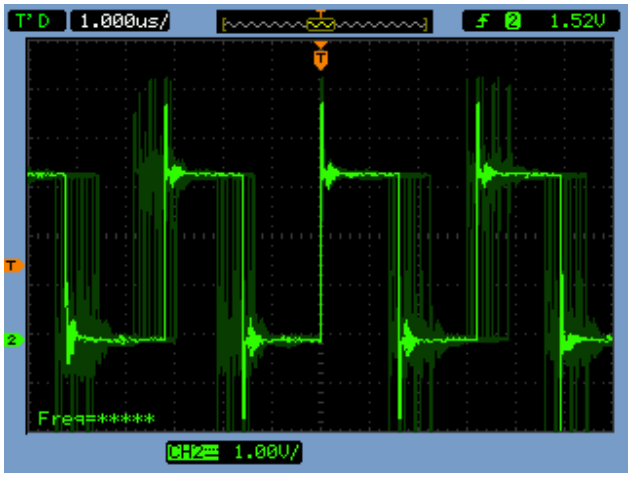


Figure 89. GPIO Forwarding – Standard Mode

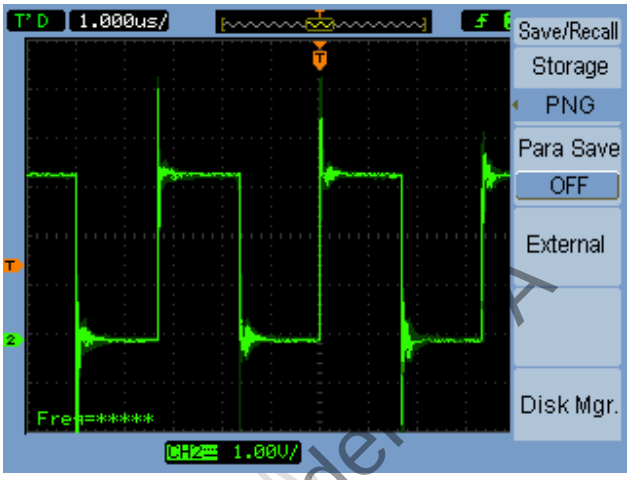


Figure 90. GPIO Forwarding – Delay Compensated Mode

The above plots show a 300kHz square wave that has been transmitted from a GMSL2 deserializer to a GMSL2 serializer. *Figure 89* shows the serializer GPIO output when configured in standard mode; *Figure 90* shows the same output with delay compensation enabled. Note the substantial reduction in jitter in *Figure 90* compared to *Figure 89*.

Note: The delay compensated mode delay value may incur a 1.7µs offset (i.e., one step value) following a link reset event.

27.3.5 Bidirectional Operation Configuration

Bidirectional operation allows GPIO inputs on each side of the link to be transmitted across the serial link. To configure the link for bidirectional function, enable the GPIO output driver and set it into open-drain mode, enable both Tx and Rx across the serial link, and set the TX_ID and RX_ID on both sides to be identical.

The following procedure configures an active-low, bidirectional GPIO scheme between a GPIO on the serializer and a GPIO on the deserializer.

1. Write GPIO_OUT_DIS = 0 to enable the output driver.
2. Set open-drain mode by writing OUT_TYPE = 0.
3. Write GPIO_TX_EN = 1 to enable the GPIO Tx.
4. Write GPIO_RX_EN = 1 to enable the GPIO Rx.
5. Set both the Tx and Rx IDs to the same value (GPIO_TX_ID = GPIO_RX_ID).

The input pull-up/pull-down resistor configuration and delay compensation can be configured as needed.

ADI Confidential for NewVision under NDA
Analog Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

27.4 Application Examples

27.4.1 Toggling GPIOs via Register Settings

GPIO pins can be manually controlled via register writes. Write to the local device to toggle local GPIO pins; write to the remote device using the control channel to toggle remote GPIO pins.

- Set GPIO_OUT_DIS = 0 and configure output driver by setting OUT_TYPE to the desired output mode (open drain or push-pull).
- Set GPIO_RX_EN = 0 to disable the GPIO receive block for the GPIO pin. This sets the GPIO to receive its value from the bitfield GPIO_OUT instead of from the value being transmitted across the GMSL2 link.
- Set GPIO_OUT to the desired value.

27.4.2 GPIO Forwarding

This is the primary mode of operation for GPIOs within GMSL2 systems. An example setup is shown below in Figure 91. In this example, GPIO0 and GPIO1 of the serializer are mapped to GPIO0 and GPIO2 of the deserializer, respectively, and GPIO1 of the deserializer is mapped to GPIO2 of the serializer.

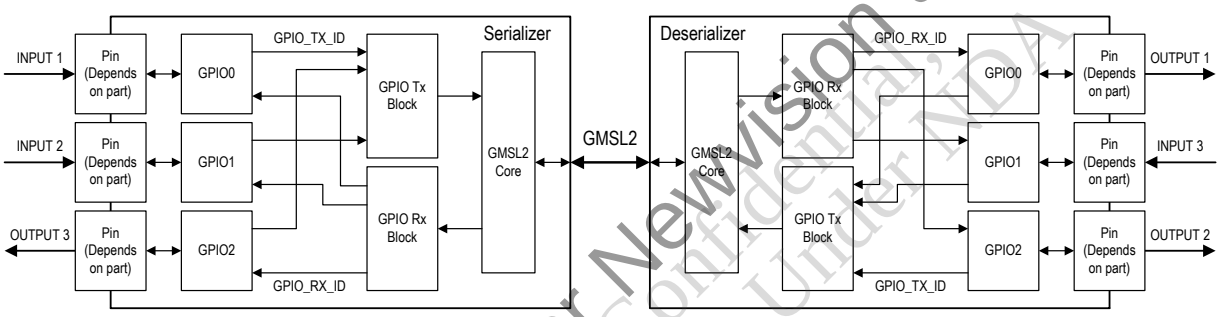


Figure 91. Example of GPIO Forwarding with Three GPIOs

27.4.2.1 GPIO Forwarding Example

Serializer Programming Procedure:

GPIO0: The serializer sets logic level.

1. GPIO_OUT_DIS = 0
2. GPIO_TX_EN = 1
3. GPIO_RX_EN = 0
4. GPIO_TX_ID [4:0] = 1

GPIO1: The logic level is applied externally.

1. GPIO_OUT_DIS = 1
2. GPIO_TX_EN = 1
3. GPIO_RX_EN = 0
4. GPIO_TX_ID [4:0] = 2

GPIO2: Tracks deserializer GPIO2.

1. GPIO_OUT_DIS = 0
2. GPIO_TX_EN = 0
3. GPIO_RX_EN = 1
4. GPIO_RX_ID [4:0] = 3
5. OUT_TYPE = 1

ADI Confidential for New Vision under NDA
Maxim Confidential
Protected by GMSL2_Customers Under NDA

Deserializer Programming Procedure:

GPIO0: Tracks serializer GPIO0.

1. GPIO_OUT_DIS = 0
2. GPIO_TX_EN = 0
3. GPIO_RX_EN = 1
4. GPIO_RX_ID [4:0] = 1

GPIO1: Tracks serializer GPIO2.

1. GPIO_OUT_DIS = 1
2. GPIO_TX_EN = 1
3. GPIO_RX_EN = 0
4. GPIO_TX_ID [4:0] = 3

GPIO2: Deserializer sets logic level.

1. GPIO_OUT_DIS = 0
2. GPIO_TX_EN = 0
3. GPIO_RX_EN = 1
4. GPIO_RX_ID [4:0] = 2

Write RESET_ONESHOT = 1 (on either serializer or deserializer) to apply reset the link and ensure GPIOs are synchronized.

The value of GPIO_IN (GPIO_A bit 3) will reflect the value detected on the input GPIO and the value of GPIO_RECVED (GPIO_C bit 6) will reflect the value received on the output GPIO.

ADI Confidential for New Vision under NDA
Analog Devices Confidential - Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

27.4.3 GPIO Broadcasting

Transitions on a single GPIO input can be mapped to multiple GPIO outputs. This can be configured by setting multiple GPIOs to the same RX_ID. The example below (Figure 92) shows a single input routed to two output GPIO pins.

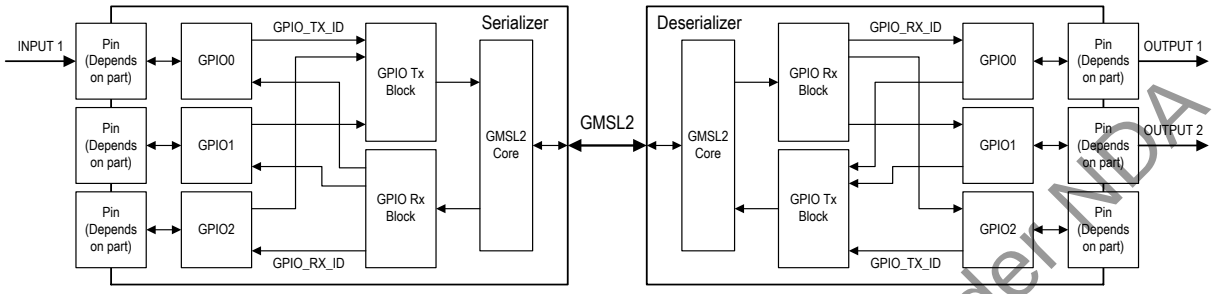


Figure 92. Example of a GPIO Input Routed to Multiple Outputs

27.4.3.1 One GPIO Input Driving Two GPIO Outputs Example

Serializer Programming Procedure:

GPIO0: The logic level is applied externally.

1. GPIO_OUT_DIS = 1
2. GPIO_TX_EN = 1
3. GPIO_RX_EN = 0
4. GPIO_TX_ID [4:0] = 1

Deserializer Programming Procedure:

GPIO0: Tracks serializer GPIO0.

1. GPIO_OUT_DIS = 0
2. GPIO_TX_EN = 0
3. GPIO_RX_EN = 1
4. GPIO_RX_ID [4:0] = 1

GPIO1: Tracks serializer GPIO1.

1. GPIO_OUT_DIS = 0
2. GPIO_TX_EN = 0
3. GPIO_RX_EN = 1
4. GPIO_RX_ID [4:0] = 1

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

27.5 Feature Availability by Device Family

All GMSL2 devices support GPIO.

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

27.6 MFP Slew Rate

GMSL2 devices' multi-function pins (MFPs) have configurable rise and fall times (slew rate). This parameter may be referred to as the I/O "speed (control)," "slew (rate)," or "edge rate" in register control bit names. Note that the MFP slew rate cannot be adjusted independently on a per-pin basis. MFPs are divided into separate speed groups; the slew rate adjustment register contains a bitfield for each group that configures the rise and fall time to all pins in the group. Refer to the device-specific datasheet for the relevant register map and MFP speed grouping details.

The MFP edge transitions must be fast enough to meet the application's requirement; however, the high-speed I/O of the GMSL link and video protocols (e.g., HDMI, MIPI, eDP/DP, RGMII, OLDI, etc.) are sensitive to coupling and crosstalk from MFP transitions. Care should be taken at a system level to prevent high edge rates and high frequencies on the MFP inputs close to these I/O. In general, the MFP pins should be configured to the slowest slew rate that allows proper function to mitigate I/O interference.

Note: Coupling refers to both inductive and capacitive coupling. Higher VDDIO supply values will increase the MFP edge rate and energy; this can introduce additional noise into the high-speed I/O.

High MFP slew rates, especially combined with high toggle frequencies, near the GMSL or high-speed video pins may adversely affect performance of the data path, including CRC errors, 9b10b code or disparity errors, reduction of link margin, and/or loss of link lock.

27.6.1 MFP Slew Rate Operation

The configurable slew rate applies to the various MFP functions differently:

1. **I2C/UART**: MFP pins operating as an I2C or UART function (i.e., control channel or pass-through) are not affected by the MFP rise/fall setting. The I2C/UART circuitry has a fixed falling-edge slew rate, and the rising-edge slew rate is determined by the external pullup resistor.
2. **Dedicated Function**: The rise and fall times of MFP pins assigned dedicated functions (e.g., SPI, I2S, RCLKOUT, or LOCK & ERR) can be adjusted by the MFP slew rate control registers.
3. **GPIO**: MFP pins operating as GPI or GPO can be adjusted by the MFP rise/fall slew rate control register.

The VDDIO supply voltage affects the I/O slew rate. The impact of the chosen VDDIO voltage must be considered when programming MFP slew rates.

Note: The register address containing the MFP slew rate may also contain the RGMII slew rate configuration in some devices. Reference the *RMII/RGMII (Ethernet)* section for additional details of this configuration.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

27.6.2 MFP Slew Rate Programming and Configuration

GMSL2 device MFPs are divided into speed groups by digital function. The slew rate adjustment register configures the rise and fall times for each MFP in the group simultaneously. The MFP slew rate can be adjusted at any time, and the changes will be applied immediately.

The MFP slew rate configuration applies to all pins in the speed group regardless of the enabled function of the pin. For example, the speed setting will be applied to a GPIO and a dedicated pin function if both are in the same MFP speed group.

The I2C/UART functions are not affected by the MFP slew rate adjustment. If an MFP is used as an I2C or UART pin, the slew rate will automatically be adjusted to meet the applicable specification.

The device VDDIO level determines the available range of the slew rate configuration options. For each VDDIO level, the MFP speed groups have four available speed options configured by two speed control bits.

The location of MFP slew rate speed control bits varies depending on the device and the available functions. Table 119 contains the register location(s) of the slew rate controls and the total number of MFP speed groups for each GMSL2 device family.

Table 119. MFP Speed Control Registers

Device Family	MFP Speed Control Register	Number of MFP Speed Groups
HDMI Serializers	CMU4	3
Advanced HDMI Serializers	GPIO_SPEED	3
DSI Serializers	CMU4	3
Advanced DSI Serializers	CMU4	3
CSI-2 Serializers	CMU4	3
Advanced CSI-2 Serializers	CMU4	3
CSI-2 Camera Deserializers	CMU4	4
CSI-2 Display Deserializers	CMU4	3
OLDI Deserializers	CMU4	2
eDP/DP Deserializers	GPIO_SPEED	3
CSI-2 Quad Deserializers	CMU4	3

Refer to the corresponding device's datasheet "Control- and Side-Channel Typical Rise and Fall Times" section for VDDIO timing details. Typical rise and fall times for GMSL2 devices are presented in Table 120.

Table 120. Typical Rise/Fall Times for GMSL2 Devices

Register Value	Rise/Fall Time	
	VDDIO=1.8V	VDDIO=3.3V
0x0	2n	1n
0x1	4n	2n
0x2	8n	4n
0x3	16n	8n

ADI Confidential for New vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

27.6.3 MFP Slew Rate Applications and Examples

27.6.3.1 Example 1. Changing the speed of MFP1 to 1ns on the Advanced HDMI Serializer

The GMSL2 Advanced HDMI Serializer MFP slew rate is configured with the GPIO_SPEED register. This register has the following mapping:

MISC: GPIO_SPEED (0x264)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSVD		GROUP_A_SPEED		GROUP_B_SPEED		GROUP_C_SPEED	

BITFIELD	BITS	DESCRIPTION	DECODE
GROUP_A_SPEED	5:4	Controls GPIO Speed Group A transition time. First value is for VDDIO = 1.8V, second value is for VDDIO = 3.3V.	0: 2ns, 1ns 1: 4ns, 2ns 2: 8ns, 4ns 3: 16ns, 8ns
GROUP_B_SPEED	3:2	Controls GPIO Speed Group B transition time. First value is for VDDIO = 1.8V, second value is for VDDIO = 3.3V.	0: 2ns, 1ns 1: 4ns, 2ns 2: 8ns, 4ns 3: 16ns, 8ns
GROUP_C_SPEED	1:0	Controls GPIO Speed Group C transition time. First value is for VDDIO = 1.8V, second value is for VDDIO = 3.3V.	0: 2ns, 1ns 1: 4ns, 2ns 2: 8ns, 4ns 3: 16ns, 8ns

The Advanced HDMI Serializer datasheet indicates that GPIO0, 1, 2, 4, 10, 11, 12, and 17 are in speed group C. This group is configured with register GROUP_C_SPEED[1:0]. Changes to this register will apply to the entirety of speed group C.

In order to set the MFP slew rate to 1ns, the device VDDIO value must be 3.3V. Writing 0b00 to register GROUP_C_SPEED[1:0] will set the slew rate to 1ns. All other groups should be set to their lowest rate (0b11) unless alternate configuration is necessary.

Example script to set the Advanced HDMI Serializer MFP1 to 1ns:

```
#Set GROUP_C_SPEED to 1ns, and set group A and B speed to their lowest speed
setting
0x90,0x0264,0x3C
```

27.6.3.2 Example 2. CMU4 Register Example Using the CSI-2 Serializer

The GMSL2 CSI-2 Serializer is an example of a GMSL2 device that uses the CMU4 register to configure the MFP slew rate. This register has the following mapping:

CMU4 (0x304)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPI_MS_SLW		SPI_LS_SLW		REFOUT_SLW		GP_SLW	

Note: Register mappings vary by device family; refer to the device-specific datasheet for register mapping details.

API Confidential for New Vision under NDA
Analog Confidential Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

28 Serial Peripheral Interface (SPI)

28.1 Overview

The serial link bridges several interfaces, including SPI, between GMSL2 devices. SPI is a serial communication interface that provides a simple connection between ICs and requires less IOs than parallel buses. Data is synchronized with the clock signal and transmitted via two unidirectional data lines. A separate control line determines when the interface is active. This simplicity allows for implementation without dedicated hardware or complex software code. This section provides setup and initialization details for multiple SPI link use cases.

Key Features:

- 4-wire Master (connects to remote peripheral) or 4-wire Slave (connects to µC/SoC).
- Remote-side SPI bus supports SPI modes 0 or 3; local-side SPI bus supports SPI mode 0.
- Device filtering on (multiple SPI interfaces with different SPI IDs) or off (point-to-point SPI interface).
- Slave Select active low or high.
- 600kHz to 25 or 50MHz SPI clock (depending on device).
- MSB first (for control commands).
- Pin or I2C control of RO and BNE input/output.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

28.1.1 GMSL2 SPI Architecture

GMSL2 enables an SPI master on one the side of the link to control an SPI peripheral on the opposite side. Functionally, the GMSL2 link does not act as a transparent bridge. On the local side, an internal SPI slave receives data from an external SPI master and transmits it across the serial link. On the remote side, the device receives the data and uses an internal SPI master to transmit the data to the external SPI slave devices. Within the serial link, this has the effect of appearing as a 4-wire SPI master controlled by a 4-wire SPI slave. Figure 93 shows a block diagram of the GMSL2 SPI interface.

Data on one side is forwarded from the transmit FIFO through the GMSL2 link and into the remote receive FIFO. This data transfer incurs a slight transmit delay. Due to this delay, the local side (connected to the µC) operates with transmit data sent first. Once data is received, the µC then reads back the received data.

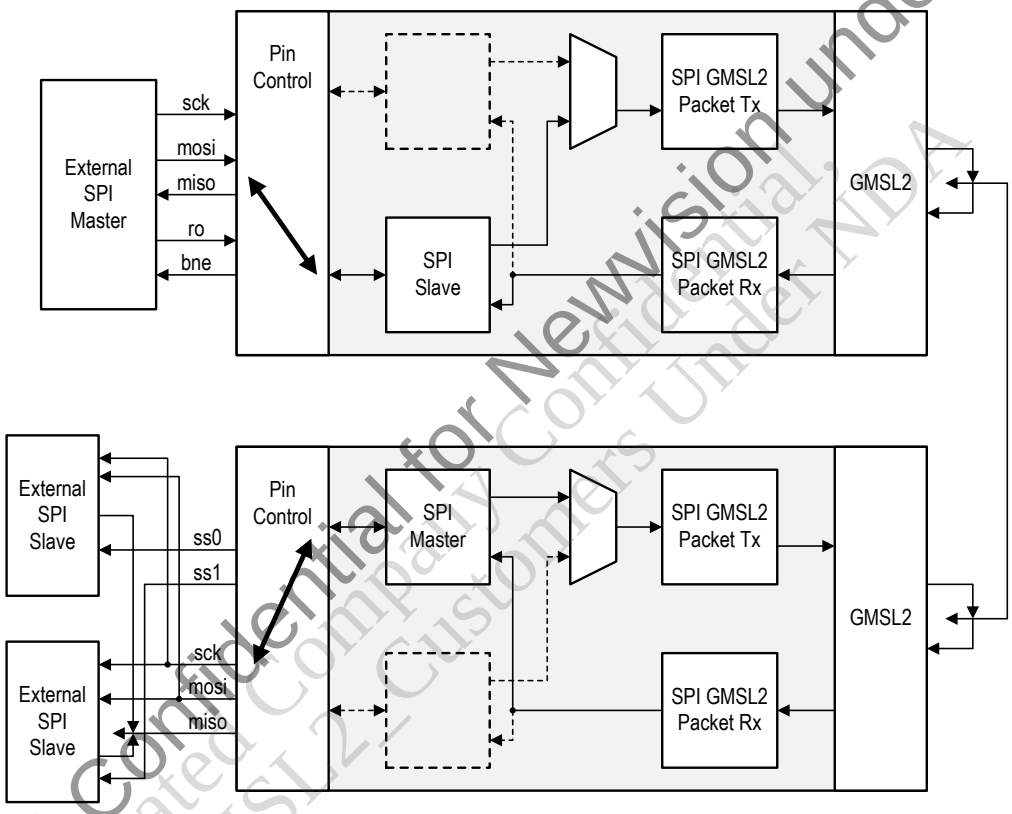


Figure 93. GMSL2 SPI Interface

ADI Confidential for Newvision under NDA
Analog Devices Confidential, under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

28.2 Operation

28.2.1 SPI Bridge

The SPI bridge has a Tx buffer used to store bytes prior to transmission on the GMSL2 link and an Rx buffer used to store bytes received from the GMSL2 link prior to being read via an SPI transaction. All SPI modes provide this buffering, which is done in both directions. Each buffer has overflow detection logic with status bits that can be read at SPI_TX_OVRFLW and SPI_RX_OVRFLW. The status bits are latching and are set with any overflow event. Status bits clear automatically upon read.

28.2.2 SPI Slave Control Bits/Pins

The SPI slave uses two bits/pins for control: Read Only (RO) and Buffer Not Empty (BNE).

Note: Refer to the latest device datasheets for SPI MFP pins. Some MFP pins may have default alternate functions that must be disabled before enabling SPI. If the SPI pins are also used as CFG pins, do not pull CFG pins until the device powers up and the CFG pins are latched.

28.2.2.1 Read Only (RO)

RO is an input bit that determines if the SPI slave is in read/command or write mode.

Write mode is enabled if RO is set low. In this mode, any data clocked into the SPI port will exit the remote SPI port. This mode is used to write data to an external peripheral (i.e., SPI slave device).

Read/command mode is enabled if RO is set high. Any data sent during this time is interpreted as control commands for SPI. Control commands are used to select an SPI port, set/clear CS/SS pins, or clock out data from the receive FIFO.

28.2.2.2 Buffer Not Empty (BNE)

BNE is an output bit that shows the receive FIFO state. BNE is low when the buffer is empty; BNE is high when there is data in the buffer. This bit is used to determine the status of the buffer for data transfers and avoiding buffer overflow. This signal will de-assert during a read operation and will re-assert if the buffer remains not empty after the read has completed.

Ensure that the buffer is empty before starting an SPI data transfer. If it is not empty, clock out the excess data until the buffer is empty (BNE = 0). BNE is also indicates the availability of a read byte on the local device. This helps in avoiding buffer overflow. Refer to the *Read Data* and *Configuration* sections below for more details.

ADI Confidential for New Revision under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_ Customers Under NDA

28.2.3 SPI Control Commands

Several control commands are used when RO is high and read/command mode is enabled.

28.2.3.1 Device Select

These commands select which GMSL device will respond based on the programmable SPI ID. These are only used in multipoint GMSL topologies and are not used in point-to-point applications (e.g., with GMSL2 quad deserializers). See the *Multiple SPI IDs* section below.

- 0xA0: Select SPI ID '00'
- 0xA1: Select SPI ID '01'
- 0xA2: Select SPI ID '10'
- 0xA3: Select SPI ID '11'

28.2.3.2 Slave Select (SS)

These commands control the remote slave select outputs. Note that actual output voltage depends on the programmed SS polarity.

- 0xA4: Assert SS1 output
- 0xA5: Assert SS2 output
- 0xA6: De-assert both SS outputs

28.2.3.3 Read Data

Sending the control byte (0xA7) during a normal buffer read allows the user to request the read of another byte from the remote side without having to toggle RO. This allows multiple data bytes to be read from the SPI slave without toggling the RO bit and requires only half as many local-side SPI byte accesses (i.e., all reads instead of alternating reads and writes).

Once there is at least a single byte to read in the local buffer (BNE = 1), the byte can be read (RO = 1), and an additional byte read request from the remote-side SPI slave can be sent (by sending 0xA7 into the MOSI). The next SPI slave read byte is ready to be read from the local buffer when BNE returns high.

When BNE returns high, the next SPI slave read byte is ready to read from the local buffer.

28.2.4 SPI Clock

The SPI clock (SCK) on local-side SPI slave is determined by the external SPI master.

The specified SPI master timing on the remote-side SPI bus is slow due to long I/O path delays. Depending on the SPI slave Clk->Q delay, operating this bus at 50MHz can be challenging. To improve performance reliability, the GMSL2 SPI interface provides a mode to use the full SCK clock period for off-chip, Clk->Q, and on-chip timing for MISO reading.

The FULL_SCK_SETUP register bit sets whether MISO is sampled after a half or full SCK period. Normal SPI timing has the slave transition on the falling clock edge and the master transition on the rising clock edge (half SCK period, FULL_SCK_SETUP = 0). If required, the GMSL2 master device can read the slave MISO data on the falling edge (full SCK period, FULL_SCK_SETUP = 1). Note that the FULL_SCK_SETUP bit has no effect on the GMSL2 slave device.

Analog Devices Confidential Distribute Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_ Customers Under NDA
ADI Confidential for New Vision under NDA

28.2.4.1 SPI SCK

The SCK on the remote-side SPI master can be set between 600KHz to 50MHz using register settings. Register SPI_4 is used for changing SCK low time and SPI_5 is used for changing SCK high time in numbers of 300MHz clocks.

Example 1: SCK = 1MHz
Write 0x96 (dec 150) to registers SPI_4 and SPI_5.
SCK = 300MHz / (150 x 2) = 1MHz

Example 2: SCK = 5MHz
Write 0x1E (dec 30) to registers SPI_4 and SPI_5.
SCK = 300MHz / (30 x 2) = 5MHz

Example 3: SCK = 42.85MHz:
Write 0x03 (dec 03) to register SPI_4 and write 0x04 (dec 04) to register SPI_5.
SCK = 300MHz / (3 + 4) = 42.85MHz

For higher SCK values, some GMSL2 device SPI MFP pins may need to be adjusted to have faster speed group settings (i.e., the rise and fall transition times for each MFP pin).

Note: GMSL2 HDMI Serializers do not have speed controls for MFP pins and no changes are required for different SCK speeds. For other GMSL2 devices, MFP speed group settings can be changed using register settings. Refer to the latest device-specific datasheet for the recommended SPI latching edge and speed group details.

Note: The maximum SPI bandwidth is SCK/8, however, there may be overhead due to BNE sampling, initiating, SPI traffic from the SoC, and/or programming overhead from the SPI slave. Actual throughput depends on the implementation and may be less than the calculated maximum bandwidth.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Subject to Copyright
Maxim Integrated Company Confidential, Subject to NDA
Provided to GMSL2_Customers Under NDA

28.2.4.2 Minimum Timing Requirements

Use register SPI_3 to guarantee that minimum timing requirements are met between the assertion of SS and the start of SCK clocks, the end of SCK clocks and the de-assertion of SS, or the time between de-assertion of SS and re-assertion of SS.

All three timing events use the same 8-bit field which defines the minimum number of 300MHz clock cycles allowed (*Table 121*).

Table 121. SPI Minimum Timing Requirements

Bit	Label	R/W	Description	Default
7:0	SPIM_SS_DLY_CLKS	R/W	Number of 300MHz clocks to delay between: 1. Assertion of SS and Start of SCK pulses 2. End of SCK pulses and De-assertion of SS 3. De-assertion of SS and Re-assertion of SS (if necessary). 4. 0xXX: Number of clock delays	00000000

For all modes of operation, control registers are available to limit the minimum and maximum GMSL2 SPI packet payload as well as set the transmit request priority. These are:

- SPI_LOC_N: This control limits the maximum packet size generated for GMSL2 transmission to (2N+1) bytes. The default value is 6'd7, which limits the packet size to 15 bytes. If this value is programmed to be larger than 7 bytes, the SPI ARQ function must be disabled to avoid ARQ buffer overflows. Reference the CRC Error Detection and ARQ Error Correction section for additional information.

- REQ_HOLD_OFF: This controls defines the minimum number of extra bytes required in the Tx buffer prior to requesting the GMSL2 link. The default value of 3'd0 requires no extra bytes in the Tx buffer, and the link request will be granted as soon as there is data to transmit. Note: All available bytes (as limited by SPI_LOC_N) will be sent when the link request has been granted.

- REQ_HOLD_OFF_TO: The control defines the timeout duration for the REQ_HOLD_OFF logic in terms of 100ns units. With a timeout, a GMSL2 request will be issued regardless of how many extra bytes are required per the REQ_HOLD_OFF control field. The default value of 8'd0 disables the timeout function.

- SPI_BASE_PRIORITY: This field defines the SPI bridge request priority for the GMSL2 link. The priority levels increase lowest to highest from 0 to 3. The priority will automatically increase by one, if possible, when the Tx buffer is over half-full. The default value is 2'd1.

ADI Confidential for New Revision under NDA
Analog Devices Confidential, Distribution Prohibited
Maxim Integrated Confidential, Distribution Prohibited
Provided to GMSL2_ Customers Under NDA

28.3 Configuration

28.3.1 Initialization

Configure SPI in the following order to initialize SPI (starting from the default values):

1. Configure SPI mode 0 or 3 on the serializer and deserializer and set SS output polarity (remote side).
2. Set the clock delay and high/low times (in number of 300MHz clocks).
3. Program the IO pin enables (BNE/RO/SS1/SS2).
4. Configure Master/Slave mode, SPI ID (if needed), and enable SPI.

28.3.2 Sending a 4-wire SPI Command (up to 15 Bytes)

Perform the following to send a full duplex command:

1. Set RO high to put the device into command mode.
2. Check BNE to ensure that the buffer is empty. If not, clock out the excess data until the buffer is empty.
3. (Optional) send 0xA0 – A3 to select which SPI Device to talk to.
4. Send 0xA4 or 0xA5 to assert SS1 or SS2 on the remote master.
5. Set RO low to put the device into write mode.
6. Write 4 data bytes on MOSI.
7. Wait until BNE is high (indicating that data has been received from the remote device), then set RO high to put the device into command mode.
8. BNE will be high for the number the bytes available in the FIFO.
9. Send 0xA6 to de-assert SS1 and SS2. Read one data byte on MISO.
10. Check if BNE is still high.
11. If BNE is still high, send 0xFF to read more bytes on MISO.
12. Set RO low to put the device into write mode.



Figure 94. GMSL2 SPI Implementation

ADI Confidential for NewVision under NDA
Maxim Confidential, Distribute only under NDA
Provided to GMSL2 Integrated Customers Under NDA

28.3.3 SPI Burst Read/Write

The SPI Tx FIFO is 16 bytes and Rx FIFO is 32 bytes. It is possible to read/write burst data (more than 16 bytes) across the GMSL link. A burst write access generates a sequence of write bytes, while a burst read access reads multiple bytes from the read buffer. The number of bytes available to read in the buffer is available in the status field SPIS_BYTE_CNT.

Buffer overflows can be avoided by tracking the number of bytes that are moving through the SPI bridge. This measurement is referred to as Bytes in Transit (BIT). The potential for buffer overflows is avoided by maintaining a maximum BIT less than the maximum buffer size (16 bytes).

At the beginning of the burst, initiate a string of write transactions until the maximum BIT has been met. Maintain the maximum BIT during the burst by initiating a new write transaction for every data byte that is read via a read transaction. At the end of the burst, perform only read transactions until the BIT is zero.

28.3.3.1 SPI Burst Write

1. Set RO.
2. Send 0xA0 (Set SPI Target = 0, optional if only one device).
3. Send 0xA4/A5 (Assert SS1/SS2).
4. Clear RO.
5. Send Cmd Byte (Read/Write and Address MS bit).
6. Send Adrs Byte.
7. Send Write Byte.
8. Set RO.
9. Wait for BNE = 1.
10. Send 0x00/Read Byte (Discard).
11. Clear RO.
12. Repeat 7-11 until all data is written.
13. Set RO.
14. Wait for BNE = 1.
15. Send 0xA6/Read Byte (Discard) (Clear SS).
16. Send 0xA6/Read Byte (Discard) (Clear SS).

28.3.3.2 SPI Burst Read

1. Set RO.
2. Send 0xA0 (Set SPI Target = 0, optional if only one device).
3. Send 0xA4/A5 (Assert SS1/SS2).
4. Clear RO.
5. Send Cmd Byte (Read/Write and Address MS bit).
6. Send Adrs Byte.
7. Set RO.
8. Wait for BNE = 1.
9. Send 0xA7/Read Byte (Read Data) (Discard first two reads, remaining are valid).
10. Repeat Steps 8–9 until all but two bytes are read.
11. Wait for BNE = 1.
12. Send 0xA6/Read Byte (Valid Data) (Clear SS).
13. Send 0xA6/Read Byte (Last Valid Data) (Clear SS).

Maxim Integrated Confidential, Distribute only to Customers Under NDA
Provided to CMSL2_ Customers Under NDA
AD Confidential for NewVision under NDA

28.3.4 Multiple SPI IDs

In point-to-point SPI applications, the GMSL devices can be set to ignore the packet IDs and accept all packets (SPI_IGNR_ID = 1). For multipoint GMSL topology (e.g., one serializer linked to two deserializers or vice-versa), each remote-side device can be assigned a different SPI ID using register writes to filter SPI packets. Four SPI IDs (i.e., A0 – A3) are available for this application.

Example 1: One serializer (MAX9295D) → Two deserializers (MAX96752)

The configuration below is used to create a SPI network using one GMSL2 serializer as an internal slave and two GMSL2 deserializer devices as internal masters to be able to control up to four slaves using Splitter Mode (Figure 95).

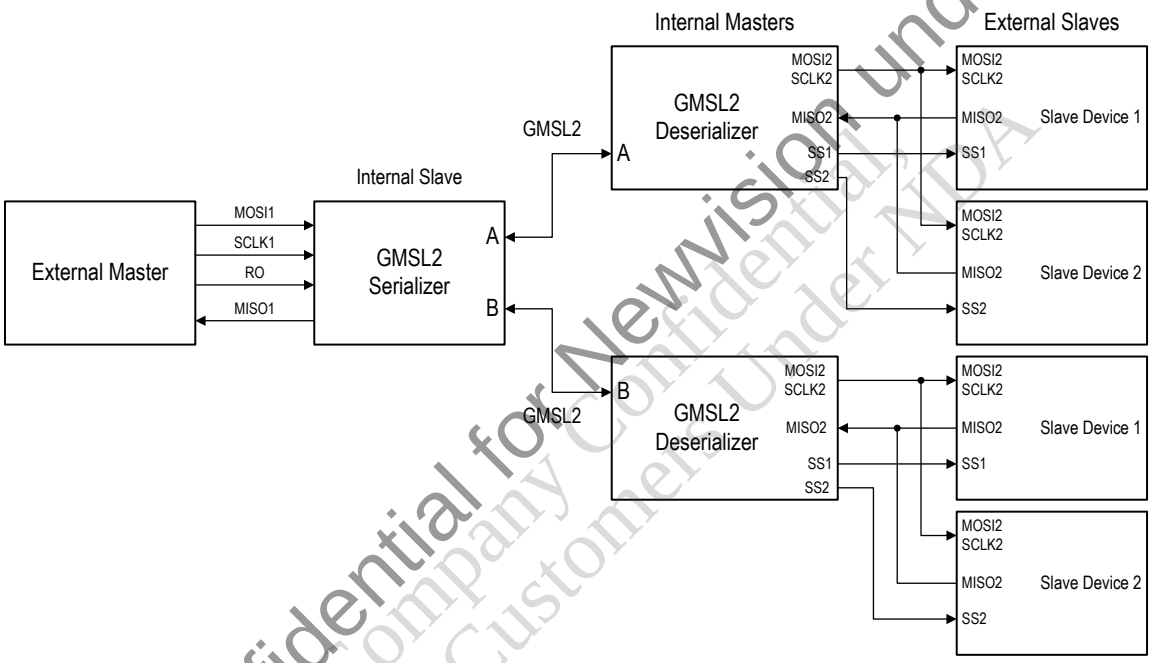


Figure 95. SPI Network: Two Deserializers and One Serializer

In this SPI network, splitter mode is enabled, and the SPI registers are configured to acknowledge the header ID to determine packet acceptance. The two MAX96752 deserializers are linked to one MAX9295D serializer and communicate with each external slave by sending command bytes of 0xA0 (link A) or 0xA1 (link B) to select the desired link and 0xA4 or 0xA5 to select the desired Slave Select (SS).

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customer Under NDA

Script:

```
#SPI 1 to 2 communication
#enable splitter mode

#Ser SLAVE
#SER
##############################SPLITTER MODE SETUP################################
#set splitter mode, auto link, and reset link
0x80,0x10,0x53
#turn off reset link
0x80,0x10,0x13
##############################SPLITTER MODE SETUP################################

###################################SPI SETUP#####################################
#SPI0
0x80,0x170,0x9
#SPI1
0x80,0x171,0x0
#SPI2
0x80,0x172,0x0
#SPI3
0x80,0x173,0x0
#SPI4
0x80,0x174,0x2C
#SPI5
0x80,0x175,0x2C
#SPI6
0x80,0x176,0x03
#SPI7
0x80,0x177,0x98

#Des MASTER
#DES dev90
#SPI mode 0 default
#SPI0
0x90,0x160,0x3
#SPI1
0x90,0x161,0x2
#SPI2
0x90,0x162,0x4
#SPI3
0x90,0x163,0x20
#SPI4
0x90,0x164,0x80
#SPI5
0x90,0x165,0x80
#SPI6
0x90,0x166,0xC
#SPI7
0x90,0x167,0x0

#Des MASTER2
#DES dev94
#SPI mode 0 default
#SPI0
0x94,0x160,0x43
#SPI1
```

ADI Confidential for Newvision under NDA
Maxim Device Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

```
0x94,0x161,0x2
#SPI2
0x94,0x162,0x4
#SPI3
0x94,0x163,0x20
#SPI4
0x94,0x164,0x80
#SPI5
0x94,0x165,0x80
#SPI6
0x94,0x166,0xC
#SPI7
0x94,0x167,0x0
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Example 2: Two serializers (MAX9295A) → One deserializer (MAX9296A)
Similarly, an SPI network using two GMSL2 serializer devices as internal masters and one GMSL2 deserializer device as an internal slave can be configured to control up to four external slaves using Reverse Splitter Mode (Figure 96).

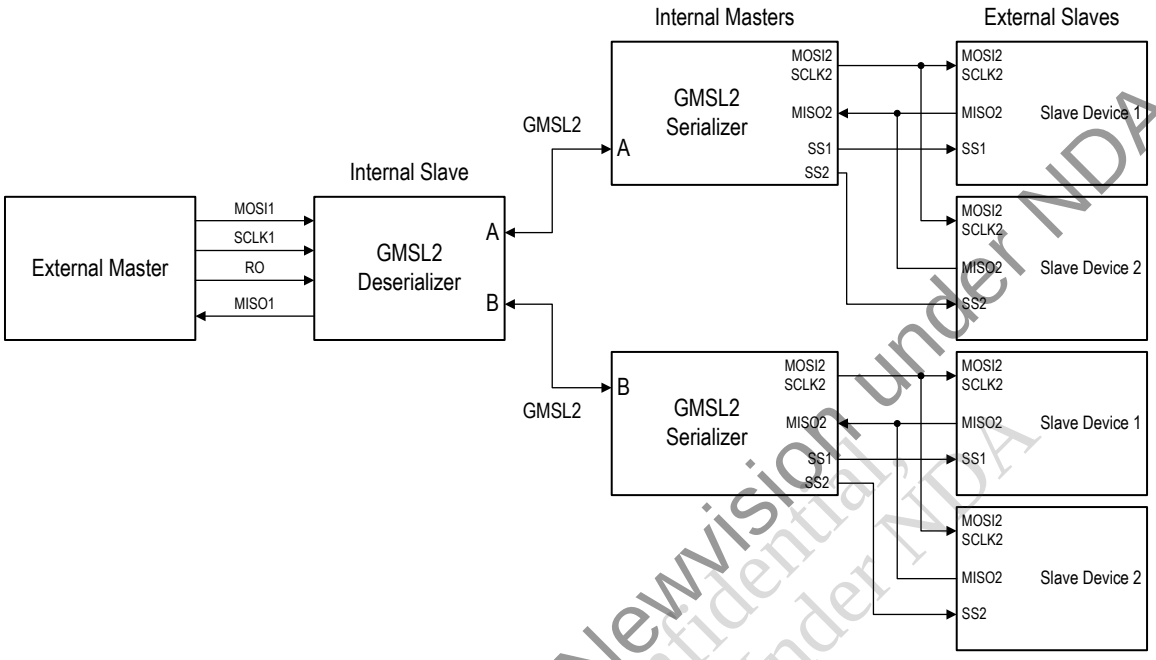


Figure 96. SPI Network: Two Serializers and One Deserializer

Script:

```
#Script for two Ser and one Des SPI
#2 x MAX9295A --> MAX9296A
#5MHz SCK output
#
#Reset Parts - Remove if not required
#0x90,0x0010,0x91
#0x80,0x0010,0x91
#disable UART pass-through on Des
0x90,0x0003,0x00
#
#0x90 deserializer is configured as µC local side
#0x80 and 0x84 serializer are configured as µC remote side
#
#enable reverse splitter mode
0x90,0x0010,0x23
0x90,0x0170,0x09
0x90,0x0176,0x03
#0x90,0x0162,0x08
#
0x80,0x0173,0x1E
0x80,0x0174,0x1E
0x80,0x0175,0x1E
0x80,0x0176,0x0C
```

Maxim Confidential
ADI Confidential for Newvision under NDA
Provided to GMSL2_Customers Under NDA
Maxim Integrated Company Confidential

```
0x80,0x0172,0x00
#To ignore the SPI ID
#0x80,0x0170,0x0B
#SPI ID = 0x01, use 0xA1 command
0x80,0x0170,0x53
#
0x84,0x0173,0x1E
0x84,0x0174,0x1E
0x84,0x0175,0x1E
0x84,0x0176,0x0C
0x84,0x0172,0x00
#To ignore the SPI ID
#0x84,0x0170,0x0B
#SPI ID = 0x00, use 0xA0 command
0x84,0x0170,0x03
```

28.3.5 GMSL2 Quad Deserializers

GMSL2 CSI-2 quad deserializers are equipped with two pass-through SPI blocks that can be used with GMSL2 serializers to bridge the SPI interface through the serial link. With SPI pass-through operation, an SPI master on one side of the GMSL2 link can control SPI peripherals on the opposite side. GMSL2 CSI-2 quad deserializer SPI ports do not support broadcasting and are intended for point-to-point communication.

There are two SPI ports that can interface with two separate GMSL2 serializers via different links. The GMSL2 link must be specified via registers (SPI_0_LINK_SELECT and SPI_1_LINK_SELECT). Ports cannot be switched between links on the fly; the SPI engine must first be disabled before switching links.

Reference the GMSL2 CSI-2 Quad Deserializer section for more information and additional device support.

ADI Confidential for Review only under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

28.3.6 Typical Application

In the example script below, it is assumed that external SPI master (µC) is connected to a GMSL2 deserializer (e.g., MAX9296A), and the peripheral SPI devices are connected to a GMSL2 serializer (e.g., MAX9295A). Reference Figure 97 for the block diagram of this application.

Note: If the µC is connected to the GMSL2 serializer, interchange the register settings for the serializer and the deserializer as indicated in the script.

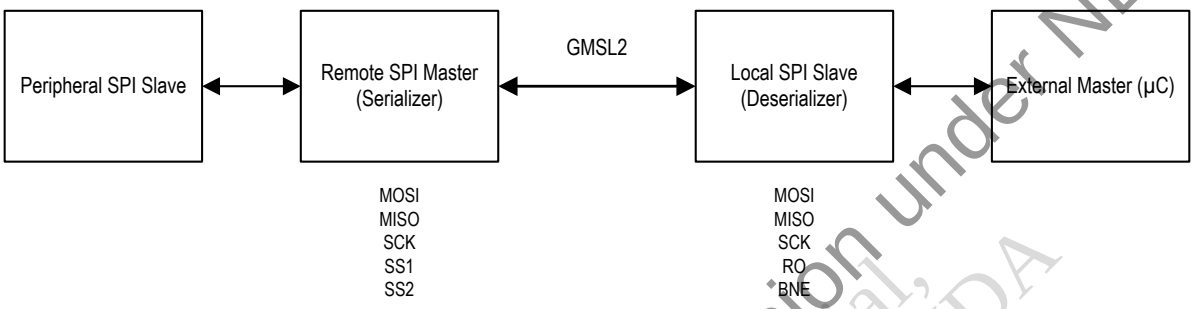


Figure 97. Typical SPI Application

Script:

```
#Script for MAX9295A/96A SPI
#
#disable UART pass-through on Des
0x90,0x0003,0x03
#
#0x90 deserializer is configured as µC local side
#0x80 serializer is configured as µC remote side
#swap only below registers 0x80 and 0x90 if µC is on Ser side
#
#SCK = 5MHz
0x80,0x0173,0x1E
0x80,0x0174,0x1E
0x80,0x0175,0x1E
#SS1 and SS2 enable
0x80,0x0176,0x0C
#enable RO and BNE
0x90,0x0176,0x03
#SS is active low
0x80,0x0172,0x00
#enable SPI, ignore SPI ID and internal SPI slave
0x90,0x0170,0x09
#enable SPI, ignore SPI ID and internal SPI master
0x80,0x0170,0x0B
```

Maxim Integrated Confidential, Distribution Limited
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Vision under NDA

28.4 Feature Availability by Device Family

All GMSL2 devices support SPI. Refer to device-specific datasheets for SPI MFP pins.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

29 RMII/RGMII (Ethernet)

29.1 Overview

RMII (Reduced Media-Independent Interface) and RGMII (Reduced Gigabit Media-Independent Interface) are full-duplex, bidirectional interfaces that serve as a connection between the Ethernet MAC and the Ethernet PHY. GMSL2 devices with RMII/RGMII transport capabilities include Ethernet packets with the packetized data sent in serial link transmissions (Figure 98).

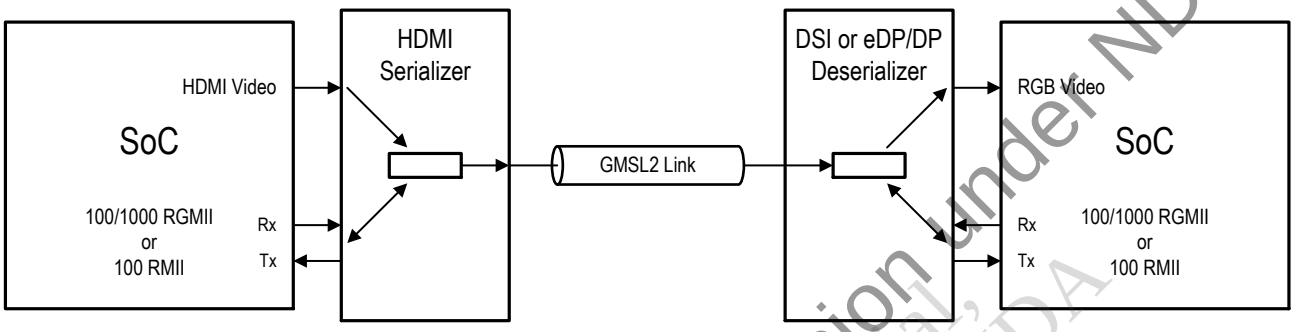


Figure 98. Video and Ethernet Packets Transmitted in a GMSL2 Link System

29.1.1 Features

- 100Mbps or 1000Mbps Ethernet.
- RMII and RGMII interface supported.
- Bi-directional pass-through interface over the side channel.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Disclosure
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

29.2 Operation

29.2.1 RMII vs. RGMII

RMII and RGMII differ on the physical layer by the number of Tx and Rx pins, reference clock frequency, and maximum data rate. The Ethernet standard specifies that RMII can operate at 10Mbps and 100Mbps, whereas RGMII can operate at 10Mbps, 100Mbps, or 1000Mbps (i.e., double data rate). Table 122 provides the reference clock frequency and Tx/Rx data signals for various RMII/RGMII configurations.

Note: GMSL2 products with RMII and RGMII are not specified for operation at the 10Mbps rate.

Table 122. RMII/RGMII Rate, Ref Clock Frequency, and Tx/Rx signals

Interface/Rate	Ref Clock Freq	Tx and Rx data signals	Notes
RMII 10Mbps	50MHz	2	Do not use
RMII 100Mbps	50MHz	2	
RGMII 10Mbps	2.5MHz	4	Do not use
RGMII 100Mbps	25MHz	4	
RGMII 1000Mbps	125MHz	4 (double data rate)	

GMSL2 devices use a single Ethernet block design for both RMII and RGMII, and the two interface configurations share pins. Table 123 contains a RGMII-to-RMII pin cross reference.

Table 123. RGMII-to-RMII Pin Cross Reference

RGMII	I/O	RMII
RG_RX_CTL	Input	RM_CRS_DV
RG_RD0	Input	RM_RD0
RG_RD1	Input	RM_RD1
RG_RD2	Input	RM_RX_ER
RG_RD3	Input	Not Used
RG_RXC	Input	RM_REF_CLK_IN
RG_TXC	Output	Not Used
RG_TX_CTL	Output	RM_TX_EN
RG_TD0	Output	RM_TD0
RG_TD1	Output	RM_TD1
RG_TD2	Output	RM_TX_ER
RG_TD3	Output	Not Used
RG_REF_CLK	Output	RM_REF_CLK
RG_MDC	Output	RM_MDC
RG_MDIO	In/Out	RM_MDIO

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution allowed under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_ Customers Under NDA

29.2.2 GMSL2 Bandwidth

RMII/RGMII shares GMSL2 link bandwidth with forward channel video and other side channel data. Typical GMSL2 devices have a forward serial bit rate of 3Gbps or 6Gbps and a reverse serial bit rate of 187.5Mbps or 1.5Gbps (see the GMSL2 Link Basics section for more information). Full-duplex Ethernet rates up to 1000Mbps are supported; however, maximum Ethernet bandwidth may not be available in all configurations depending on application and the bandwidth consumption of other channels. The majority of serial link bandwidth is consumed by forward channel video. Generally, as video resolution increases, available bandwidth for side channels (including Ethernet) decreases (Figure 99). Reference GMSL2 Link Bandwidth Consumption from Side Channels (RGMII) for additional information.

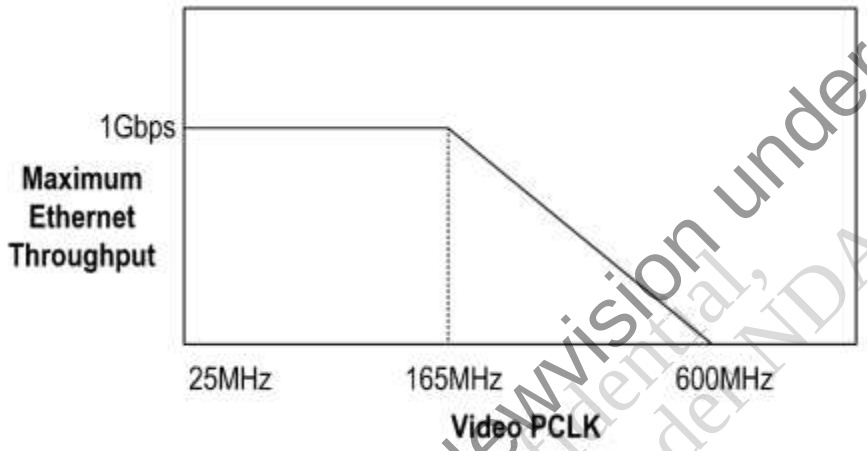


Figure 99. Relationship Between Ethernet Throughput and the Video Pixel Clock

29.2.2.1 Display Stream Compression

Display Stream Compression (DSC) can be enabled to help achieve full operational speeds of Ethernet transmissions when available bandwidth is limited. DSC provides a fixed 3:1, visually lossless compression of the video stream transmitted over the serial link.

The results of lab tests comparing various pixel clock values at RGMII link rates of 100Mbps and 1000Mbps with and without DSC are presented below in Table 124. These are provided as a guide to aid in system planning. These figures are presented as reference and will be further affected by other side channel bandwidth consumption and/or feature use.

Note: Ethernet traffic comprises burst transmissions. Due to the nature of the GMSL2 PHY packetizer and scheduler, the maximum serial link bandwidth consumption due to Ethernet burst transmissions cannot be accurately calculated.

Table 124. Lab testing results of Pixel Clock vs. RMII/RGMII rate

PCLK (Max)	RGMII Link Rate	Resolution	DSC	FEC	Other features (SPI, I2S, HDCP, etc.)
201MHz	100Mbps	1888x1584@60	N	N	None
201MHz	100Mbps	1888x1584@60	N	N	HDCP
135MHz	1000Mbps	1624x1000@60	N	N	None
135MHz	1000Mbps	1624x1000@60	N	N	HDCP

ADI Confidential for Newvision under NDA

575MHz	100Mbps	4096x2260@60	Y	Y	None
325MHz	1000Mbps	2400x2160@60	Y	Y	None
325MHz	1000Mbps	2400x2160@60	Y	Y	GMSL2 HDCP 1.4

29.2.3 SoC or PHY Connection

GMSL2 devices use RMII/RGMII pin naming conventions typical for connections to Ethernet PHYs: Rx connections mate with Rx connections and Tx connections mate with Tx connections (Figure 100).

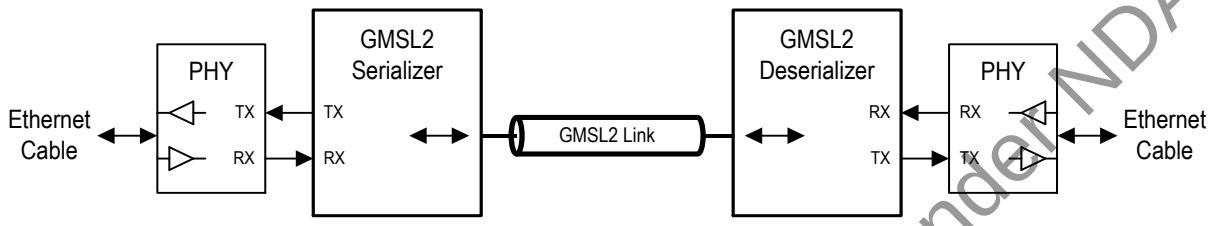


Figure 100. GMSL2 Device Connections to Ethernet PHY

When GMSL2 devices are mated with a MAC/SoC, the connection names are typically the opposite of those used with PHY connections (*Figure 101*).

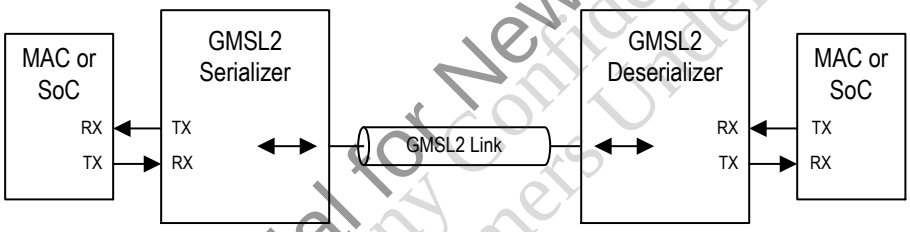


Figure 101. GMSL2 Device Connections to MAC/SoC

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute Only to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

29.3 RMII/RGMII Configuration

29.3.1 Clock-to-Data Skew

Historically, the RMII/RGMII clock has transitioned coincident with the transition of data signals (Figure 102), and a timing delay was added with either the PCB trace or timing adjustment(s) within the receiver device to center-align the clock to the data. For system flexibility, GMSL2 devices with RMII/RGMII capability incorporate clock-to-data timing adjustment at either the Tx or Rx pins to center the clock in the valid data period to provide adequate setup and hold times (Figure 103). To maintain proper timing, the clock-to-data delay must be added at either the SoC or the GMSL2 device at both sides of the GMSL2 link. This is controlled in GMSL2 devices with the ADD_TX_DELAY and ADD_RX_DELAY registers.

Note: RMII/RGMII clock-to-data skew is adjusted independently at either side of the GMSL2 link (i.e., timing adjustments are not propagated through the serial link).

Default edge-aligned RMII timing:

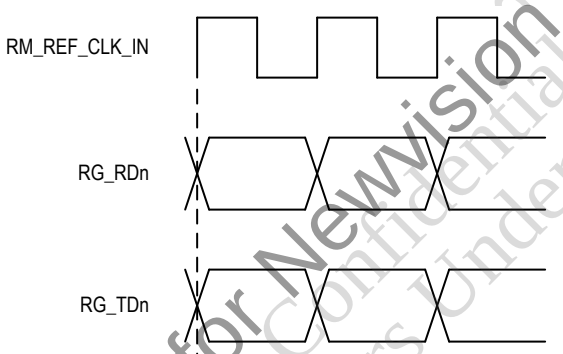


Figure 102. RMII Timings with Edge-Aligned Data when ADD_RX_DELAY = 0 and ADD_TX_DELAY = 0

Center-aligned RMII timing expected by GMSL2 devices:

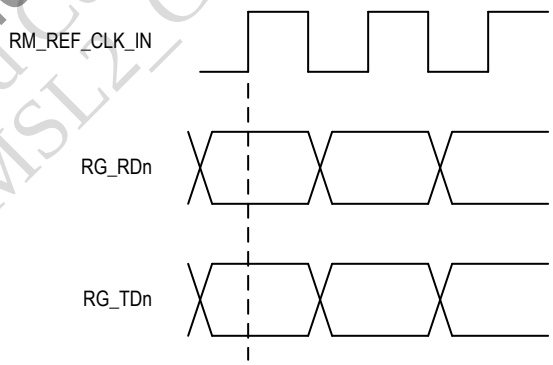


Figure 103. RMII Timings with Edge-Aligned Data when ADD_RX_DELAY = 1 and ADD_TX_DELAY = 1

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_ Customers Under NDA

The following process (*Figure 104*) details the necessary timing adjustments for a GMSL2 device that is receiving edge-aligned clock data when connected to an Ethernet PHY requiring a clock center-aligned to the data:

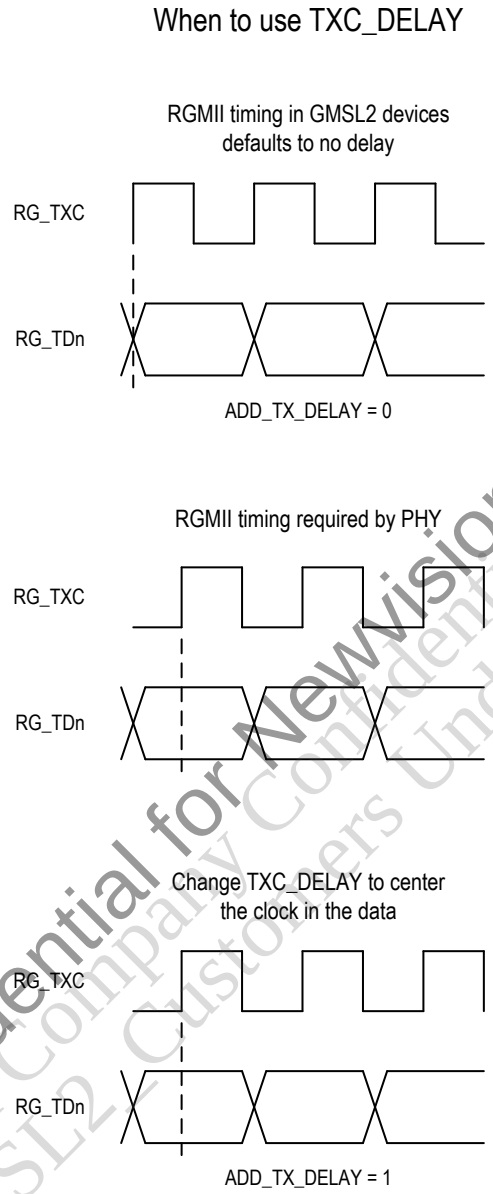


Figure 104. TXC_DELAY Usage

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

29.3.2 RMII Hardware Configuration

RMII 100Mbps is a single data rate interface that uses a common transmit and receive clock, operates at 50MHz, and uses two Tx and two Rx data signals.

GMSL2 devices use the following data signals:

RMII Transmit:

- RM_TD0
- RM_TD1
- RM_TX_EN
- RM_TX_ER
- TX_CLK: In PHY applications, the PHY TX_CLK can be generated from the PHY using RMII_CLKOUT (Figure 105) or from the GMSL2 device using RM_REF_CLK (Figure 106). For MAC/SoC applications, the GMSL2 RM_REF_CLK_IN and SoC RMII_REFCLK may be generated by the GMSL2 RM_REF_CLK (Figure 107).

RMII Receive:

- RM_RD0
- RM_RD1
- RM_CRS_DV
- RM_RX_ER
- RM_REF_CLK_IN

The GMSL2 RMII/RGMII adapter board is configured as shown in Figure 105 (reference Table 123).

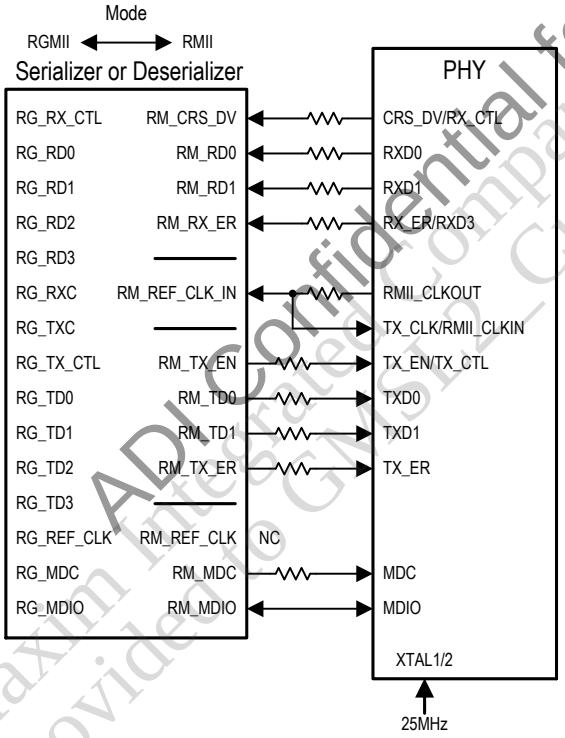


Figure 105. RMII Hardware Configuration 1 (PHY)

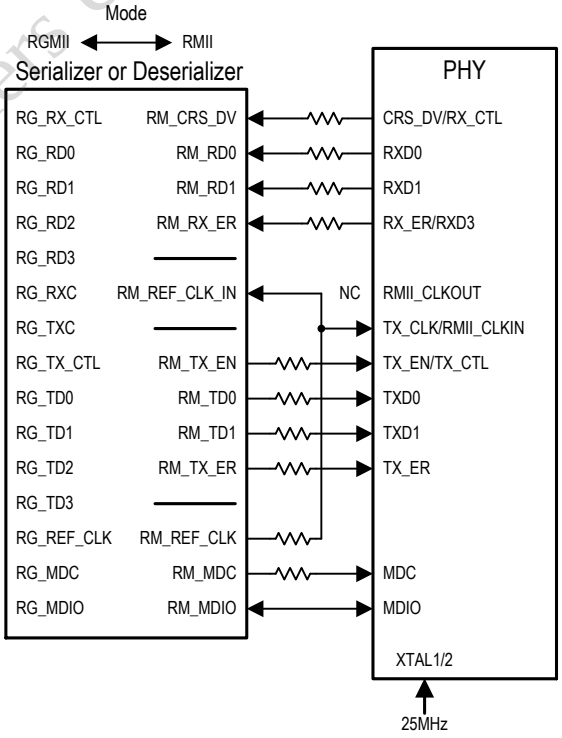


Figure 106. RMII Hardware Configuration 2 (PHY)

Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA
ADI Confidential for Newvision under NDA
Do Not Distribute or Make Public

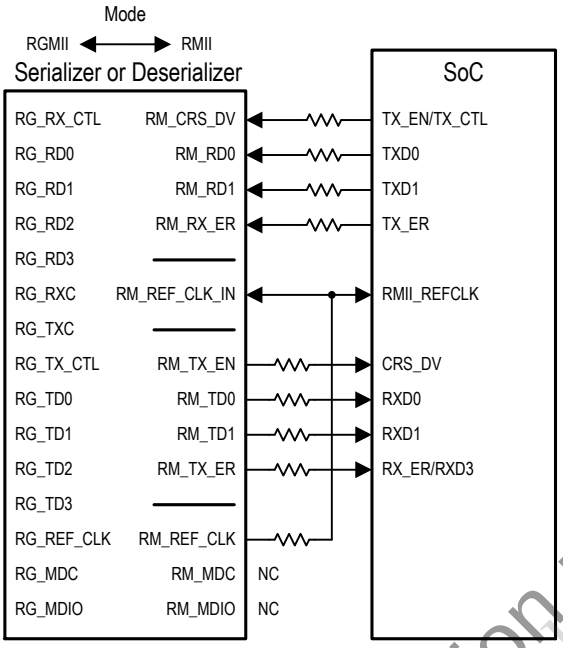


Figure 107. RMII Hardware Configuration 3 (MAC/SoC)

Note: The signals in all configurations utilize series termination resistors that will reduce ringing on these normally unterminated transmission lines if edge rates are high.

29.3.3 RMII Register Configuration

RMII is an optional feature for GMSL2 devices and must be configured and enabled via internal registers. Most required configuration is in registers RGMII_0 through RGMII_17. Refer to the tables below for details.

The GMSL2 RMII circuit block diagram with associated registers is presented in Figure 108.

Note: GMSL2 serializer/deserializer combinations do not offer flow control. Both sides of the serial link must be set to the same RMII speed (100Mbps).

The fastest RMII speed (100Mbps) is compatible with both reverse channel rates (i.e., 1.5Gbps and 187Mbps). Any RMII configuration should be compatible with any GMSL2 speed setting if system bandwidth usage provides sufficient margin.

The RMII Configuration Script section provides an example configuration script with recommended starting values.

Analog confidential
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Vision under NDA

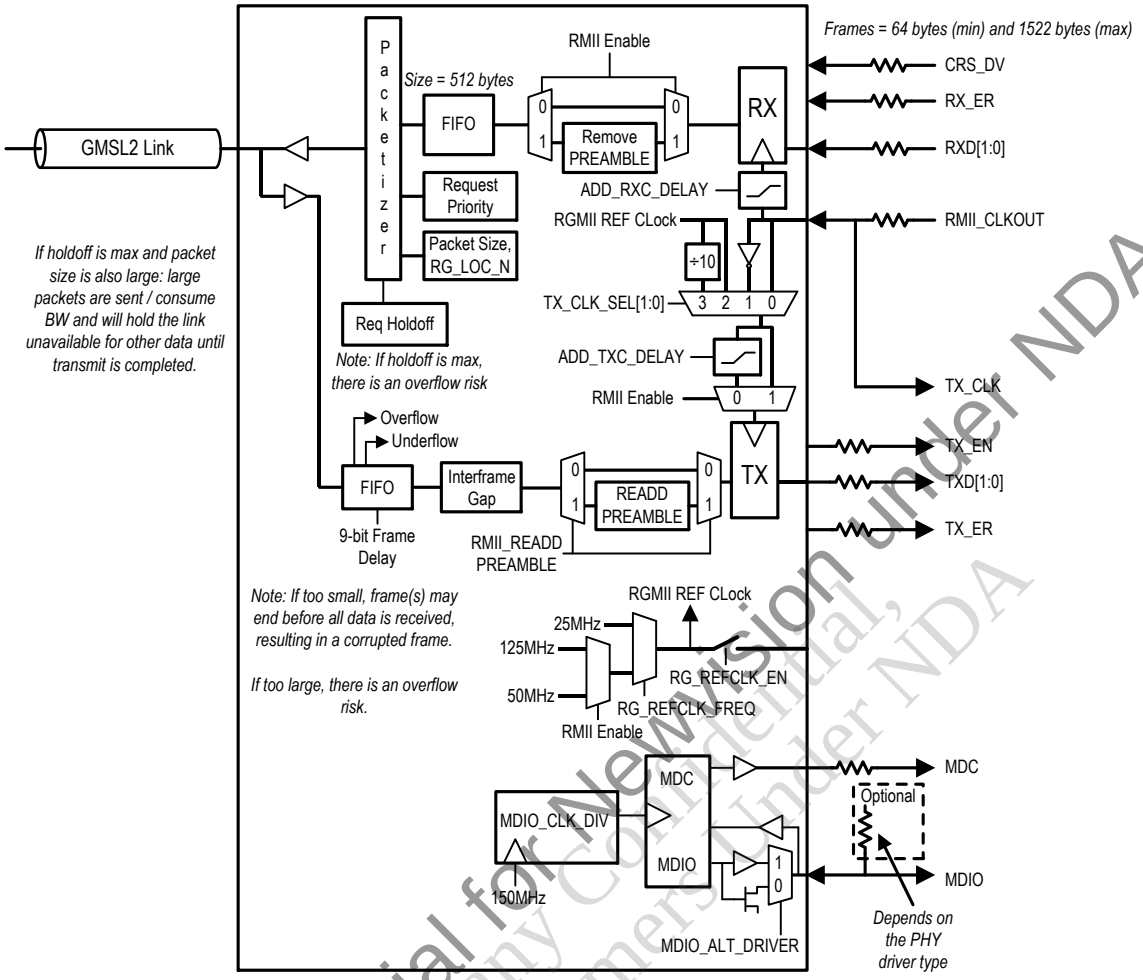


Figure 108. RMII Block Diagram

29.3.3.1 General Setup (Speed, Clock, Enable, Timing)

The following section contains important configuration registers, error flags, and operating parameters for RMII. An example configuration script is provided in the RMII Configuration Script subsection.

29.3.3.1.1 RMII/RGMII Speed Setting

The Ethernet speed is set in GMSL2 devices in the RGMII_0 register. The related reference clock divider is set in the RGMII_2 register (Table 125).

Table 125. RMII/RGMII Speed Setting Registers

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_0	RG_SPEED [1:0]	5:4	10: 1000Mbps 01: 100Mbps
RGMII:RGMII_2	RG_REFCLK_FREQ	6	1: 125MHz (50Mhz in RMII mode) 0: 25MHz

Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA
ADI Confidential for New Version under NDA

29.3.3.1.2 RMII Clock Selection

The clock is configured according to the RMII Hardware Configuration (Table 126). A block diagram of the RMII clock circuits is presented in Figure 109.

Table 126. RMII/RGMII Clock Configuration Registers

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_2	RG_REFCLK_EN	6	1: Output enabled
RGMII:RGMII_12	TX_CLK_SEL [1:0]	5:4	0: RXC 1: ~RXC (Used to meet RMII output timing) 2: RGMII Ref Clock (Used for cross-connected RGMII interfaces in 100/1000Mb mode)

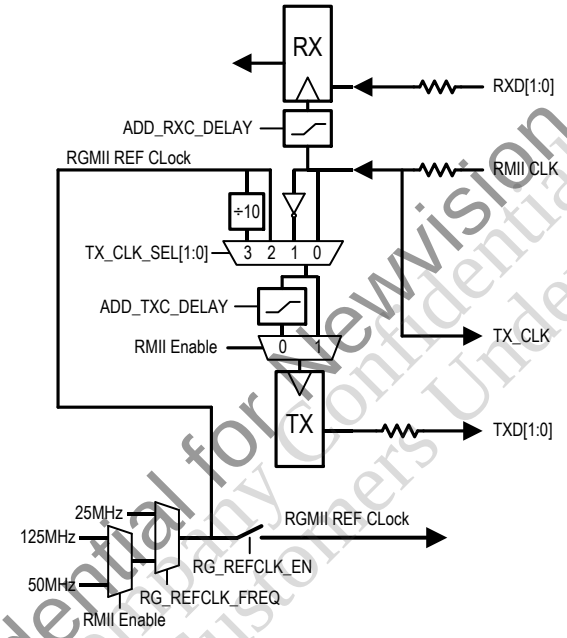


Figure 109. RMII Clock Circuit Block Diagram

29.3.3.1.3 RMII/RGMII Enable

The RGMII block is enabled with the RG_EN bit, and the mode is set with the RMII_MODE bit (Table 127).

Table 127. RMII/RGMII Enable Registers

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_0	RG_EN	1	0: Disable RMII/RGMII circuitry 1: Enable RMII/RGMII circuitry
RGMII:RGMII_6	RMII_MODE	5	0: Enable RGMII; Disable RMII 1: Enable RMII; Disable RGMII

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA

29.3.3.1.4 RMII Timing

Configure register RGMII:RGMII_15 to 0x80.

29.3.3.1.5 RMII Configuration Script

The following example script configures the serializer and deserializer in a GMSL2 system for RMII 100Mbps operation.

```
# 170 is Speed (100M), hold in reset
0x80,0x170,0x11
0x90,0x180,0x11
# 173 is SerDes RGMII Frame Config, Dynamic IFG=Enabled
0x80,0x173,0x0E
0x90,0x183,0x0E
# 174 is Frame Delay Low Byte (Default is 0x00)
0x80,0x174,0x40
0x90,0x184,0x40
# 172/182 is Ref Clock config
# Bit 6: H-125MHz or 50MHz, L=25MHz
# RMII is 50MHz
0x80,0x172,0x43
0x90,0x182,0x43
# 171 Packet size and local ID
0x80,0x171,0x1C
0x90,0x181,0xE0
# RGMII 17C is RXC and TXC Delay = 0, TX_CLK_SEL = 1, Loopback = 0
0x80,0x17C,0x50
# RMII 18C is RXC and TXC Delay = 0, TX_CLK_SEL = 2, Loopback = 0
0x90,0x18C,0x50
# 17D is REQ_HOLD_OFF_RGMII
0x80,0x17D,0x06
0x90,0x18D,0x0E
# 175 is MDIO Clock Divider, 100 clocks high, 100 clocks low
0x80,0x175,0x64
0x90,0x185,0x64
# 176 is Phy Index, RMII Enable, Serializer is RMII, Deserializer is RMII
# Bit 5: L-RGMII, H-RMII
0x80,0x176,0x20
0x90,0x186,0x20
# 177 is RMII READD Preamble and SFD
0x80,0x177,0x20
0x90,0x187,0x20
# 17F is RMII PHY adjust
# RGMII_15
0x80,0x17F,0x80
# 17D is RMII PHY adjust for devices with new registers assigned to 0x17D-0x17F
# RGMII_AC_1 (note: different register name on deserializer)
0x90,0x17D,0x80
# 170 is Speed (100M), enable RG block, duplex, release reset
0x80,0x170,0x17
0x90,0x180,0x17
```

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Distribute only under NDA
Provided to GMSL2_Customers Under NDA

29.3.3.2 Sending Ethernet Data Across the GMSL2 Link

The RMII configuration removes the Ethernet preamble. The GMSL2 packetizer evaluates the packet size and the holdoff and request priority settings to determine when data is transmitted across the serial link (Figure 110).

If the total bandwidth consumption of a given serial link system configuration does not exceed 90% of the maximum limit, the default Ethernet packet settings can typically be used. However, if bandwidth usage approaches the maximum limits, the settings may need to be adjusted to ensure proper Ethernet and serial link operation (Table 128).

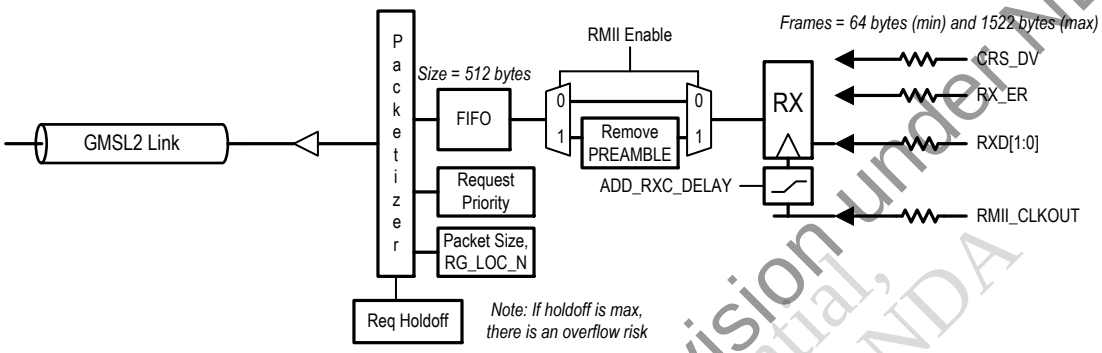


Figure 110. Sending Ethernet Data Across the GMSL2 Link

Table 128. RMII/RGMII Packet Configuration (Tx)

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_1	RG_LOC_N	7:2	Sets the packet size.
RGMII:RGMII_13	REQ_HOLD_OFF_RGMII	5:0	Hold off GMSL2 request until this number of bytes are received on the RMII/RGMII port
RGMII:RGMII_3	RG_EARLY_REQ_EN	2	0: Do not allow GMSL2 requests 1: Allow GMSL2 request
RGMII:RGMII_2	RG_BASE_PRIO	1:0	Starting GMSL2 request priority advances by one (if available) if Tx buffer is over half full. 00: Priority 1 (high) 01: Priority 2 10: Priority 3 11: Priority 4

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

29.3.3.2.1 Preamble

RMII mode removes the preamble on the transmit side of the GMSL2 link to conserve bandwidth.

29.3.3.2.2 Packet size

The packet size is the number of bytes included in the packet sent across the serial link. Proper adjustment of this setting prevents overflow or underflow conditions. Ethernet bandwidth consumption can be reduced by decreasing the size of the packet. The reduction in packet size allows Ethernet data packets to be inserted between gaps in video data.

29.3.3.2.3 Request Holdoff

Request holdoff determines how many frame bytes accumulate in the GMSL2 FIFO/buffer before a request to send across the serial link. FIFO overflow may result if the request holdoff is set too high. Ethernet frames can be up to 1522 bytes; however, the GMSL2 FIFO size is only 512 bytes. Larger frames must thus be broken up to be transferred across the serial link.

29.3.3.2.4 Early Request

Early request allows GMSL2 request(s) during a serial link transmission prior to a complete frame being received if there are enough new bytes available.

29.3.3.2.5 Priority

The base priority setting informs the GMSL2 packetizer how to prioritize RMII data relative to other data sent across the serial link. If the serial link experiences bandwidth congestion, the priority setting will automatically downgrade. In normal circumstances, the priority will remain at the set level.

29.3.3.2.6 Available Error Flags

Data is stored in the GMSL2 FIFO block prior to transmission across the serial link. RG_TX_OVRFLW is a status flag indicating an overflow condition in the GMSL2 Tx buffer (Table 129). It is cleared on read. If the overflow condition persists, it is likely that there is not enough available bandwidth to transmit the data at the specified rate.

Table 129. RMII/RGMII Tx Overflow Status Register

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_3	RG_TX_OVRFLW	5	0: No GMSL2 Tx overflow 1: GMSL2 Tx overflow

ADI Confidential for NDA vision under
Analog Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

29.3.3.3 Receiving Ethernet Data Across the GMSL2 Link

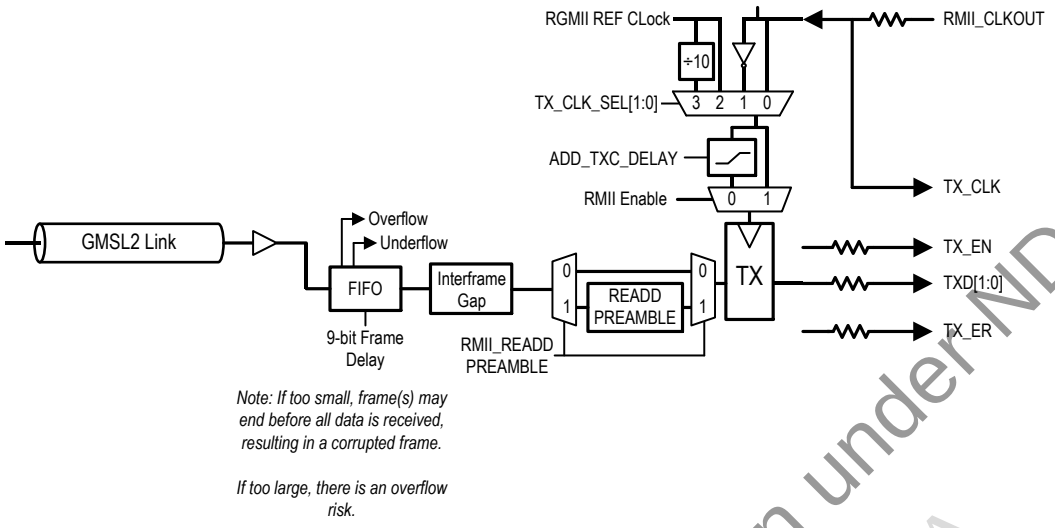


Figure 111. Receiving Ethernet Data Across the GMSL2 Link

29.3.3.3.1 Frame Delay

The frame delay setting in the in the Rx FIFO is the most critical parameter for throughput. The FIFO frame delay is set so that the FIFO is not emptied before the full Ethernet frame is transmitted across the serial link. If the FIFO empties before the full frame is sent, the receiving SoC will typically declare it an incomplete frame and discard it. This is often caused by an extremely low throughput. For proper operation, it is recommended to keep this value at 0x040.

The 9-bit frame delay parameter is contained in two registers (*Table 130*).

Table 130. RMII/RGMII Rx Frame Delay

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_3	RG_FRAME_DELAY_H	7	MSB value of frame delay
RGMII:RGMII_4	RG_FRAME_DELAY_L	7:0	Least significant 8-bits of frame delay

29.3.3.3.2 Interframe Gap

An interframe gap value can be assigned to accommodate small differences in rates at either end of the serial link and prevent possible data loss. Interframe gap values of 4, 8, 12, or 16 bytes may be specified. Alternatively, a Dynamic Interframe Gap may be enabled. This value varies without user intervention. The default interframe gap value is 12 bytes. See Table 131 for configuration details.

The RGMII clock may incur a slight timing discrepancy during transmission from the Rx input to the Tx output across the serial link. In normal use cases, the interframe gaps between Ethernet frames prevent buffer overflows and maintain packet delivery. In extended applications utilizing the full RGMII bandwidth, the ppm clock differences may result in buffer overflow and cause transmission failures. To avoid buffer overflows, additional gap time may be added to the output to accommodate a slower receive side. Enabling the Dynamic Interframe Gap reduces the interframe gap period from 12 to 11 if the buffer is at risk of overflow. This dynamic adjustment of the interframe gap uses occasional reduction to compensate the ppm clock difference.

Analog Confidential for review under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

Table 131. RMII/RGMII Interframe Gap

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_3	RG_INTER_FRM_GAP	1:0	Minimum number of interframe symbols. 00: 4 01: 8 10: 12 11: 16
RGMII:RGMII_3	RG_DYNAMIC_IFG	3	0: Dynamic Interframe Gap disabled 1: Dynamic Interframe Gap enabled

29.3.3.3.3 ReAdd Preamble

The RMII preamble is removed from the Ethernet frame at the Tx side of the link to conserve GMSL2 link bandwidth; it must be re-added on the Rx side of the GMSL2 link (Table 132).

Table 132. RMII ReAdd Preamble at GMSL2 Rx

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_7	RMII_READD_PREAMBLE	5	0: Do not add 1-byte preamble to start of frame 1: Add 1-byte preamble to start of frame (required if Tx side of link is RMII)

29.3.3.3.4 Available Error Flags

The Rx overflow error indicates that the FIFO at the receiving end of the GMSL2 link has overflowed and that data will be corrupted. The frame underflow error indicates that the FIFO emptied before the Frame was complete. Both registers are cleared on read. See Table 133 for details.

Table 133. RMII/RGMII Rx Overflow and Underflow Status Registers

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_3	RG_RX_OVRFLW	4	0: No GMSL2 Rx overflow 1: GMSL2 Rx overflow
RGMII:RGMII_3	RG_FRAME_UF	6	0: No buffer underflow 1: Buffer underflow

ADI Confidential for New Version under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

29.3.3.4 RGMII to RMII Example Configuration Scripts

The following tables contain example configuration scripts to enable RMII functionality on GMSL2 serializers (Table 134) and deserializers (Table 135). Note that both devices in a GMSL2 link system must be configured for the same mode.

Table 134. RMII/RGMII Configuration Registers for Serializer

Step	Description	Register Name	Register Write
1	Disable RGMII circuits	RGMII:RGMII_0	RG_EN = 0
2	Enable Dynamic Interframe Gap and Early Request	RGMII:RGMII_3	RG_DYNAMIC_IFG = 1 RG_EARLY_REQ_EN = 1
3	Set lower 8 bits of Frame Delay to 64 (0x40)	RGMII:RGMII_4	RG_FRAME_DELAY [7:0] = 0x40
4	Set RGMII Base Priority to High and Set Reference Clock to 50MHz	RGMII:RGMII_2	RG_BASE_PRIO[1:0] = 3 RG_REFCLK_FREQ = 1
5	Set packet size to value 3 (2N+1)	RGMII:RGMII_1	RG_LOC_N[5:0] = 3
6	Set Tx Clock Select to ~RXC and Enabled MDIO Push-Pull Driver	RGMII:RGMII_12	TX_CLK_SEL[1:0] = 0x01 MDIO_ALT_DRIVER = 1
7	Set Request Holdoff to 6	RGMII:RGMII_13	REQ_HOLD_OFF_RGMII[5:0] = 0x06
8	Set MDIO Clock Divider to 100 clocks high, 100 clocks low	RGMII:RGMII_5	MDIO_CLK_DIV[7:0] = 0x64
9	Set RGMII Block to RMII Mode	RGMII:RGMII_6	RMII_MODE = 1
10	Enable Re-Add Preamble	RGMII:RGMII_7	RMII_READD_PREAMBLE = 1
11	Set RMII DDR Bypass	RGMII:RGMII_15	DDR_BYPASS = 1
12	Enable RGMII circuits	RGMII:RGMII_0	RG_EN = 1

Table 135. RMII/RGMII Configuration Registers for Deserializer

Step	Description	Register Name	Register Write
1	Disable RGMII circuits	RGMII:RGMII_0	RG_EN = 0
2	Enable Dynamic Interframe Gap and Early Request	RGMII:RGMII_3	RG_DYNAMIC_IFG = 1 RG_EARLY_REQ_EN = 1
3	Set lower 8 bits of Frame Delay to 64 (0x40)	RGMII:RGMII_4	RG_FRAME_DELAY [7:0] = 0x40
4	Set RGMII Base Priority to High and Set Reference Clock to 50MHz	RGMII:RGMII_2	RG_BASE_PRIO[1:0] = 3 RG_REFCLK_FREQ = 1
5	Set packet size to value 49 (2N+1)	RGMII:RGMII_1	RG_LOC_N[5:0] = 49
6	Set Tx Clock Select to ~RXC and Enabled MDIO Push-Pull Driver	RGMII:RGMII_12	TX_CLK_SEL[1:0] = 0x01 MDIO_ALT_DRIVER = 1
7	Set Request Holdoff to 14	RGMII:RGMII_13	REQ_HOLD_OFF_RGMII[5:0] = 0x0E
8	Set MDIO Clock Divider to 100 clocks high, 100 clocks low	RGMII:RGMII_5	MDIO_CLK_DIV[7:0] = 0x64
9	Set RGMII Block to RMII Mode	RGMII:RGMII_6	RMII_MODE = 1
10	Enable Re-Add Preamble	RGMII:RGMII_7	RMII_READD_PREAMBLE = 1
11	Set RMII DDR Bypass	RGMII:RGMII_15	DDR_BYPASS = 1
12	Enable RGMII circuits	RGMII:RGMII_0	RG_EN = 1

ADI Confidential for New Version under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

29.3.3.5 RMII/RGMII TX Signal Rate Control

The RGMII interface allows for adjustment of the rise time of the transmit signals. Two 2-bit parameters form an array of rise time settings at either allowable VDDIO voltage.

Table 136 contains the approximate rise times for the following signals: RG_TXC, RG_TX_CTL, RG_TD[3:0], RM_TX_ER, RM_TX_ERR, RM_TX_EN, RM_TD[3:2], with 10pF load, 10%–80%.

Table 136. RMII/RGMII Tx Interface Rise Time Settings

RG_T* rise time in (nS), 1.8V/3.3V	HI_DRIVE=00	HI_DRIVE=01	HI_DRIVE=10	HI_DRIVE=11
HS_SPEED[1:0] = 00	0.829/0.466	0.670/0.386	0.559/0.386	0.494/0.287
HS_SPEED[1:0] = 01	0.881/0.472	0.734/0.391	0.623/0.320	0.560/0.290
HS_SPEED[1:0] = 10	0.901/0.474	0.768/0.386	0.667/0.321	0.606/0.291
HS_SPEED[1:0] = 11	0.950/0.479	0.836/0.391	0.743/0.325	0.675/0.295

Table 136 default settings:

- HI_DRIVE = 01
- HS_SPEED = 00
- Which is: 0.670nS if 1.8V or 0.386nS if 3.3V

Rise time programming registers vary by device family (Table 137).

Table 137. Rise Time Programming Registers by Device Family

Device Family	MISC: RGTX_SPEED
HDMI Serializers	Bits [1:0]: RGTX_SPEED Bits [3:2]: RGTX_STRENGTH
eDP/DP Deserializers	Bits [1:0]: RGTX_SPEED Bits [3:2]: RGTX_HI_DRIVE
CSI-2 Deserializers	Bits [1:0]: RGMII_TX_SPEED Bits [3:2]: RGMII_TX_HI_DRIVE

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

29.3.4 RGMII Configuration

29.3.4.1 RGMII Hardware Configuration

RGMII hardware configuration adds three pins to the RMII configuration. Use the RGMII-to-RMII cross reference (Table 123) to determine pin use and note the differences when connecting to a SoC or PHY. RGMII and RMII configurations are similar: reference the RMII Hardware Configuration and RMII Register Configuration sections before proceeding with the sections below.

Figure 112 shows the RGMII block diagram; Figure 113 shows the typical RGMII to SoC configuration.

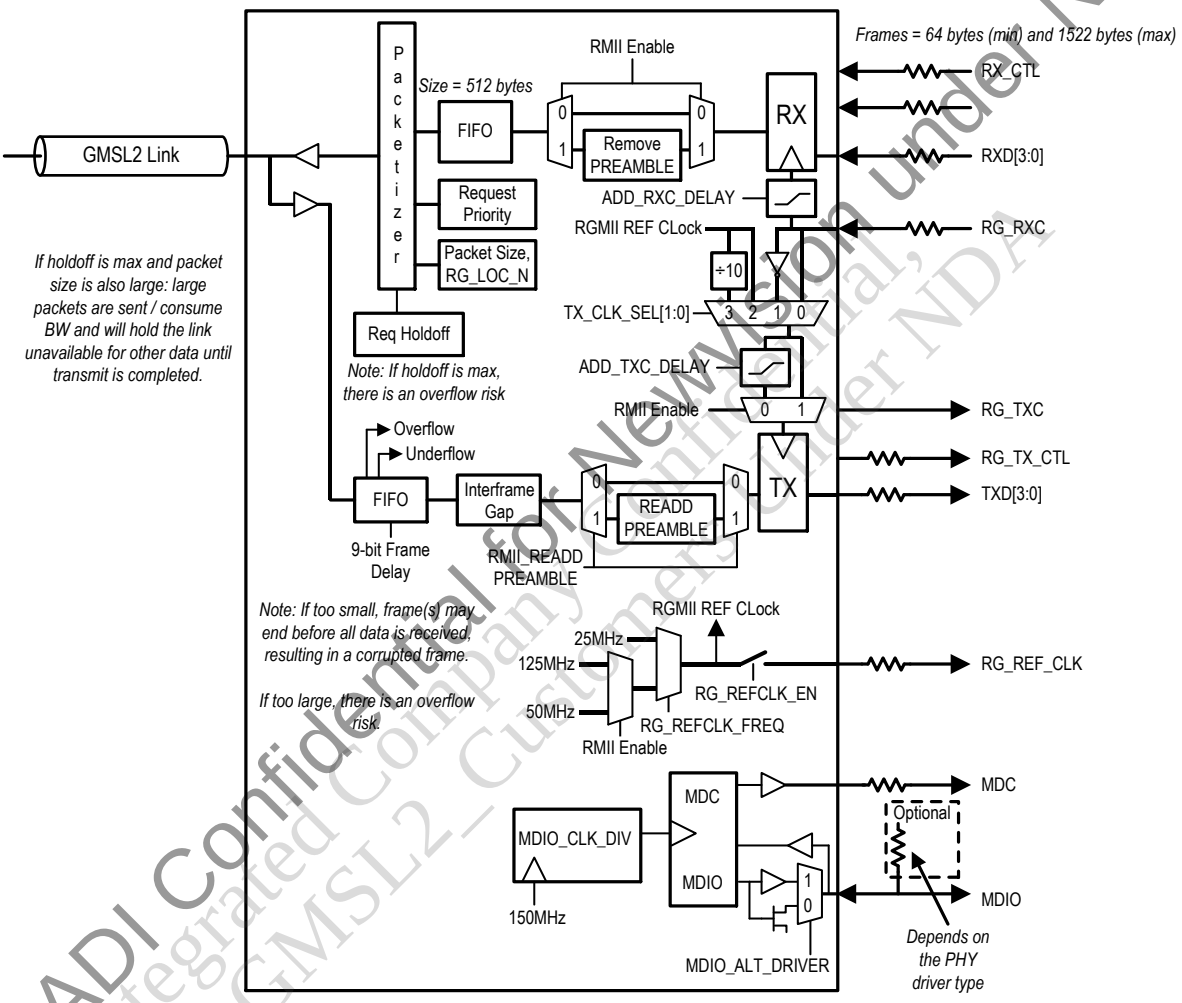


Figure 112. RGMII Block Diagram

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers under NDA

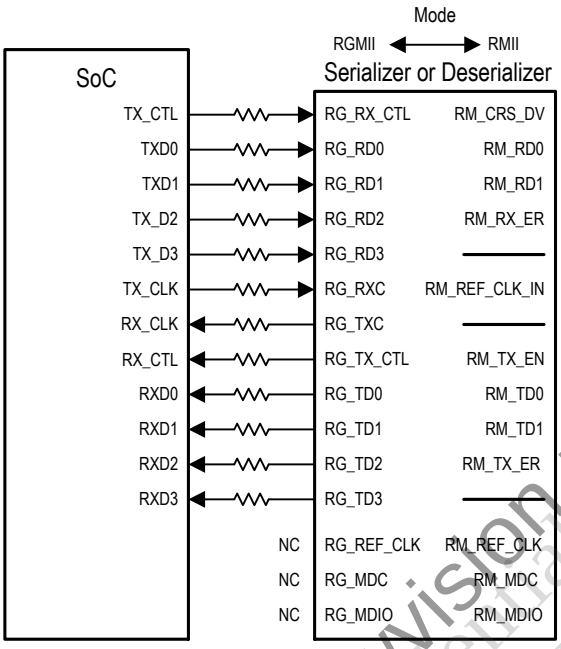


Figure 113. RGMII Hardware Configuration (SoC)

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

29.3.4.2 RGMII Register Configuration

RGMII GMSL2 register configuration uses similar settings as the RMII Register Configuration; however, there are important differences described below.

29.3.4.2.1 RGMII Mode

Set RMII_MODE to 0 to enable RGMII and set RMII_READD_PREAMBLE to 0 since the RMII preamble does not need to be added (Table 138).

Table 138. RGMII Mode Enable Registers

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_6	RMII_MODE	5	0: Enable RGMII; Disable RMII 1: Enable RMII; Disable RGMII
RGMII:RGMII_7	RMII_READD_PREAMBLE	5	0: Do not add 1-byte preamble to start of frame 1: Add 1-byte preamble to start of frame (required if Tx side of link is RMII)

29.3.4.2.2 RGMII Speed

GMSL2 RGMII devices can operate at 100Mbps or 1000Mbps. The RGMII speed is set with the RG_SPEED register. If the reference clock is required by the system design, it must be configured to the correct reference clock frequency with the RG_REFCLK_FREQ register and enabled with RG_REFCLK_EN (Table 139).

Reference clock frequency settings:

- RGMII 100Mbps = 25MHz
- RGMII 1000Mbps = 125MHz

Table 139. RGMII Speed Settings

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_0	RG_SPEED	5:4	01: 100Mbps 10: 1000Mbps
RGMII:RGMII_2	RG_REFCLK_FREQ	6	0: 25MHz 1: 125MHz
RGMII:RGMII_2	RG_REFCLK_EN	7	0: RGMII reference clock output disabled 1: RGMII reference clock output enabled

Note: Use of RG_REFCLK_EN is dependent on how the SoC is implemented and the system design. This bitfield may be used by the mating device but is not required.

29.3.4.2.3 RGMII Clock Selection

Select the RGMII clock source by setting TX_CLK_SEL = 0b10. Configure register RGMII:RGMII_15 to 0x00 (default).

Note: GMSL2 serializer/deserializer combinations do not offer flow control. Both sides of the serial link must be set to the same RGMII speed (100Mbps or 1000Mbps).

ADI Confidential for Newvision under NDA
Analog Devices Confidential Digikey to Maxim Integrated Company Confidential Provided to GMSL2_Customers Under NDA

29.3.4.2.4 RGMII Clock-to-Data Skew

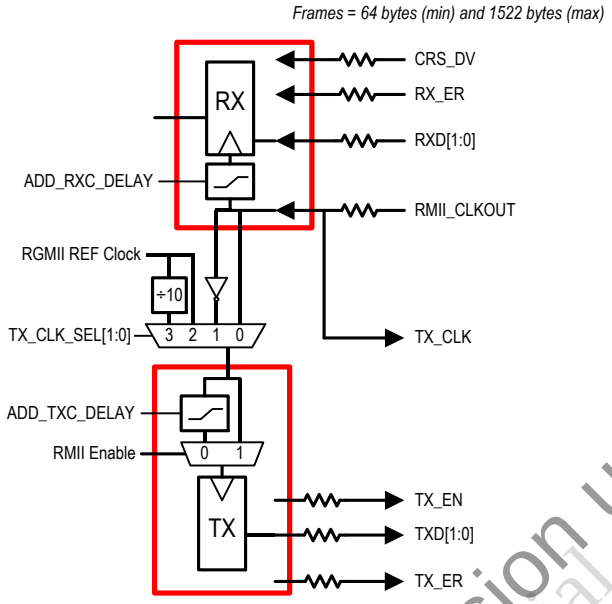


Figure 114. Clock-to-Data Skew

The clock-to-data output skew (RGMII Specification: TskewT) and the clock-to-data input skew (RGMII Specification TskewR) are key timing parameters. Figure 114 contains a partial block diagram that shows the blocks impacted by the clock-to-data skew. To accommodate different timing combinations of RGMII 2.0 or earlier versions, timing delays of ~1.8nS may be added at either the Tx output or Rx input of the GMSL2 device to ensure proper setup and hold times (Figure 115 and Figure 116). See Table 140 for register information.

Table 140. RGMII Clock-to-Data Skew Delay

Register Name	Bitfield	Bits	Decode
RGMII:RGMII_12	ADD_TXC_DELAY	0	Add ~1.8nS to Tx Clock
RGMII:RGMII_12	ADD_RXC_DELAY	1	Add ~1.8nS to Rx Clock

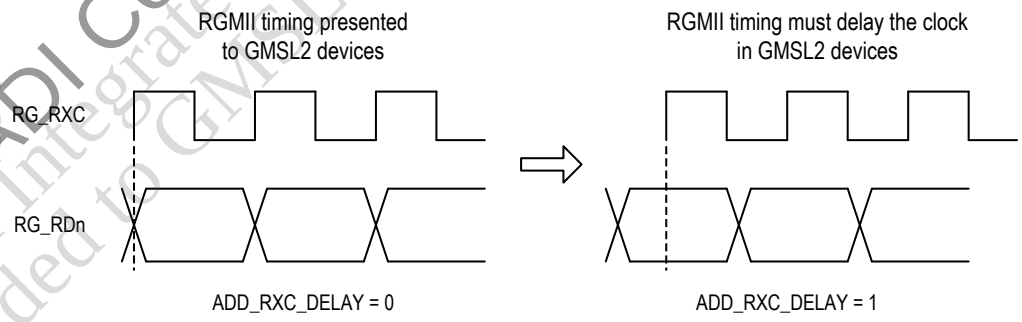


Figure 115. RXC_Delay

ADI Confidential for New Revision under NDA
Analog Devices Confidential, Distributed Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers_Under NDA

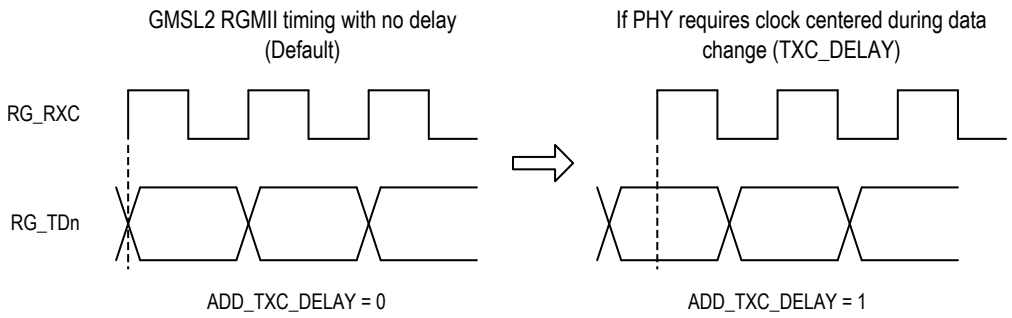


Figure 116. TXC_Delay

29.3.4.2.5 RGMII Configuration Scripts

Table 141. 1000Mbps RGMII Configuration Registers for Serializer

Step	Description	Register Name	Register Write
1	Disable RGMII circuits	RGMII:RGMII_0	RG_EN = 0
2	Enable Dynamic Interframe Gap and Early Request	RGMII:RGMII_3	RG_DYNAMIC_IFG = 1 RG_EARLY_REQ_EN = 1
3	Set lower 8 bits of Frame Delay to 64 (0x40)	RGMII:RGMII_4	RG_FRAME_DELAY [7:0] = 0x40
4	Set RGMII Base Priority to High and Set Reference Clock to 125MHz	RGMII:RGMII_2	RG_BASE_PRIO[1:0] = 3 RG_REFCLK_FREQ = 1
5	Set packet size to value 5 (2N+1)	RGMII:RGMII_1	RG_LOC_N[5:0] = 5
6	Set Tx Clock Select to RGMII and Enable MDIO Push-Pull Driver	RGMII:RGMII_12	TX_CLK_SEL[1:0] = ‘b10 MDIO_ALT_DRIVER = 1
7	Set Request Holdoff to 6	RGMII:RGMII_13	REQ_HOLD_OFF_RGMII[5:0] = 0x06 (Ser)
8	Set MDIO Clock Divider to 100 clocks high, 100 clocks low	RGMII:RGMII_5	MDIO_CLK_DIV[7:0] = 0x64
9	Disable RMII Mode	RGMII:RGMII_6	RMII_MODE = 0
10	Disable Re-Add Preamble	RGMII:RGMII_7	RMII_READD_PREAMBLE = 0
11	Disable RMII DDR Bypass	RGMII:RGMII_15	DDR_BYPASS = 0
12	Enable RGMII circuits and set RG speed to 1000Mbps	RGMII:RGMII_0	RG_EN = 1 RG_SPEED[1:0] = ‘b10

ADI Confidential for New Revision under NDA
Analog Devices Confidential Only
Maxim Integrated Company Confidential NDA
Provided to GMSL2_ Customers Under NDA

Table 142. 1000Mbps RGMII Configuration Registers for Deserializer

Step	Description	Register Name	Register Write
1	Disable RGMII circuits	RGMII:RGMII_0	RG_EN = 0
2	Enable Dynamic Interframe Gap and Early Request	RGMII:RGMII_3	RG_DYNAMIC_IFG = 1 RG_EARLY_REQ_EN = 1
3	Set lower 8 bits of Frame Delay to 64 (0x40)	RGMII:RGMII_4	RG_FRAME_DELAY [7:0] = 0x40
4	Set RGMII Base Priority to High and Set Reference Clock to 125MHz	RGMII:RGMII_2	RG_BASE_PRIO[1:0] = 3 RG_REFCLK_FREQ = 1
5	Set packet size to value 49 (2N+1)	RGMII:RGMII_1	RG_LOC_N[5:0] = 49
6	Set Tx Clock Select to RGMII and Enable MDIO Push-Pull Driver	RGMII:RGMII_12	TX_CLK_SEL[1:0] = ‘b10 MDIO_ALT_DRIVER = 1
7	Set Request Holdoff to 14	RGMII:RGMII_13	REQ_HOLD_OFF_RGMII[5:0] = 0x0E
8	Set MDIO Clock Divider to 100 clocks high, 100 clocks low	RGMII:RGMII_5	MDIO_CLK_DIV[7:0] = 0x64
9	Disable RMII Mode	RGMII:RGMII_6	RMII_MODE = 0
10	Disable Re-Add Preamble	RGMII:RGMII_7	RMII_READD_PREAMBLE = 0
11	Disable RMII DDR Bypass	RGMII:RGMII_15	DDR_BYPASS = 0
12	Enable RGMII circuits and set RG speed to 1000Mbps	RGMII:RGMII_0	RG_EN = 1 RG_SPEED[1:0] = ‘b10

ADI Confidential for New Version under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

29.3.5 RGMII to RMII Mixed-Mode Configuration

GMSL2 serial link systems may incorporate both RGMII and RMII. An RGMII-GMSL2-RMII system configuration will use combined RGMII and RMII settings. If the system receives RMII and transmits RGMII, the RGMII interface will have to add the preamble/SFD (Start of Frame Detect) removed by the RMII interface by enabling bit 5 of RMII_7. The preamble does not need to be added if the system receives RGMII and transmits RMII. In this configuration, RMII_7, bit 5 should be set to 0.

Note: GMSL2 serializer/deserializer combinations do not offer flow control. Both sides of the serial link must be set to the same RMII/RGMII speed (100Mbps).

Note: The preamble is followed by the SFD (Start of Frame Detect). The SFD is separate from the preamble and is only relevant in the mixed-mode configuration.

29.3.6 MDIO

The RMII/RGMII interface management bus is a two-wire bus that uses a GMSL2-generated clock (MDC) and a single pin, bidirectional data bus (MDIO). The MDC clock frequency is determined by the 150MHz RG reference clock divided by the MDIO_CLK_DIV[7:0] register value. The clock divider is the number of clocks high and the number of clocks low. The MDIO pin can be set for either open-drain operation (default) or as a push-pull driver with the MDIO_ALT_DRIVER bit in the RMGII:RGMII_12 register (0 = open-drain; 1 = push-pull). Note that the open-drain configuration requires the use of an external pull-up resistor (2k–5k ohms). Reference Figure 117.

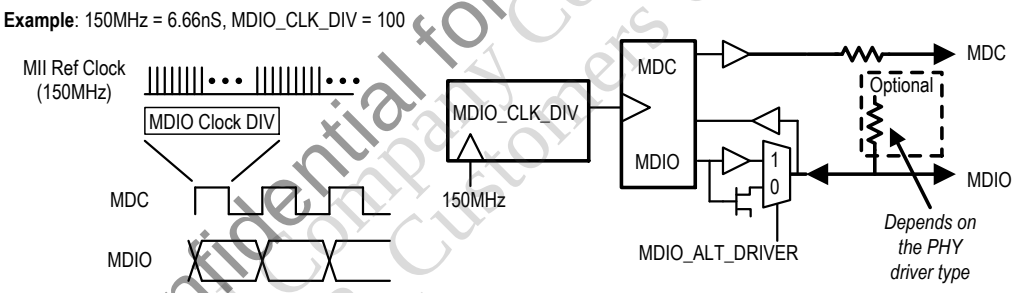


Figure 117. MDIO

The following information is required when writing to an attached device using MDIO:

- Device address,
- Register to write,
- Low byte of data, and
- High byte of data.

Confidential for New Version under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

The following tables provide MDIO write and read examples (Table 143 and Table 144).

Table 143. MDIO Write Example

Step	Description	Register Name	Register Write
1	Write the phy address to the MDIO block	RGMII:RGMII_6	MDIO_PHY_ADRS[4:0] = 'bnnnnn
2	Set the desired register address to write and set Write bit	RGMII:RGMII_7	MDIO_REG_ADRS[4:0] = 'bnnnnn MDIO_WRITE = 1
3	Enter the Low Byte data to write	RGMII:RGMII_8	MDIO_WR_DATA_L [7:0] = 0xnn
4	Enter the High Byte data to write	RGMII:RGMII_9	MDIO_WR_DATA_L [7:0] = 0xnn
5	Set MDIO_START = High to execute the write cycle	RGMII:RGMII_7	MDIO_START = 1
6	Set MDIO_START = Low to complete the write cycle	RGMII:RGMII_7	MDIO_START = 0

Table 144. MDIO Read Example

Step	Description	Register Name	Register Write
1	Write the phy address to the MDIO block	RGMII:RGMII_6	MDIO_PHY_ADRS[4:0] = 'bnnnnn
2	Set the desired register address to write and set Write bit = 0 to read	RGMII:RGMII_7	MDIO_REG_ADRS[4:0] = 'bnnnnn MDIO_WRITE = 0
3	Set MDIO_START = High to execute the read cycle	RGMII:RGMII_7	MDIO_START = 1
4	Set MDIO_START = Low to complete the write cycle	RGMII:RGMII_7	MDIO_START = 0
5	Read the Low Byte data	RGMII:RGMII_10	MDIO_RD_DATA_L [7:0] = 0xnn
6	Read the High Byte data	RGMII:RGMII_11	MDIO_RD_DATA_L [7:0] = 0xnn

29.3.7 Local Loopback

Local loopback is available in both RGMII and RMII modes (Figure 118). To enable local loopback, write RGMII:RGMII_12, bit 2, TEST_LCL_LOOPBACK = 1. Note that if RMII mode is configured, the preamble will not be present in loopback frame.

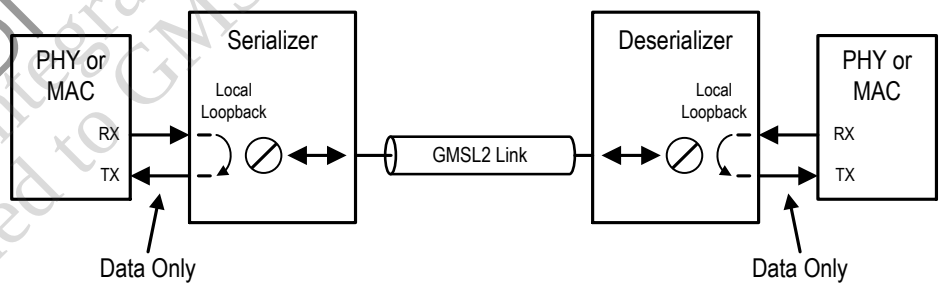


Figure 118. Local Loopback

ADI Confidential for New Version under NDA
Analog Devices Confidential, A Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

29.3.8 Remote Loopback

Remote loopback is available in both RGMII and RMII modes (Figure 119). To enable remote loopback, write RGMII:RGMII_12, bit 3, TEST_RMT_LOOPBACK = 1. Note that if RMII mode is configured, the preamble will appear in the frame.

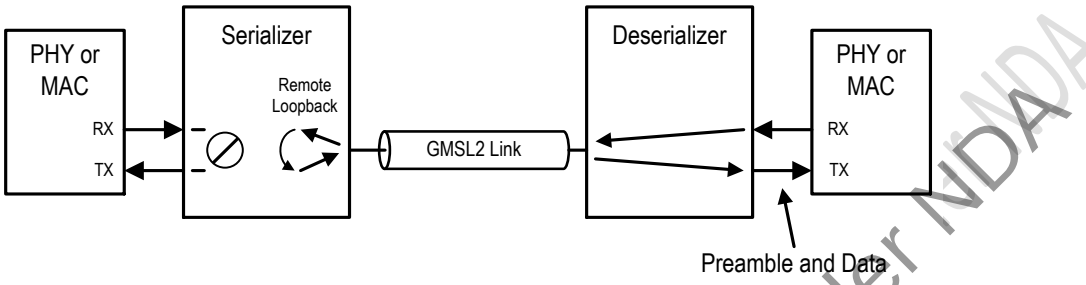


Figure 119. Remote Loopback

29.3.9 EtherSAM (Y.1564) Testing

To ensure compliant operation of Ethernet over GMSL2, ADI has tested GMSL2 devices at specific video resolutions relative to Pixel Clock (PCLK), Ethernet speeds, and with additional features to determine the maximum capability (Figure 120). Table 124 shows successful operational PCLK values at RGMII rates of 100Mbps and 1000Mbps with and without Data Stream Compression (DSC) enabled.

The test vehicle for this Ethernet testing is EtherSAM (Y.1564) operating on an Ethernet test set (EXFO MAX-860). With video running over the GMSL2 link at the specified resolution, the EtherSAM test generates TCP and UDP data at ramp, burst, and highest rate throughput.

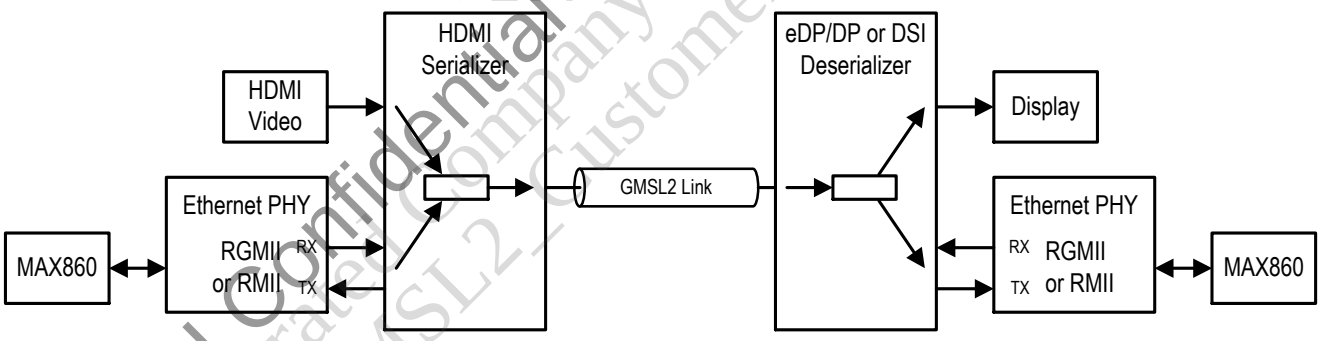


Figure 120. EtherSAM (Y.1564) Testing

29.3.9.1 Service Configuration and Service Performance Tests

The two types of tests are Service Configuration (ramp testing) and Service Performance (throughput). Maximum throughput is shown here for a 100Mbps configuration where throughput is divided between UDP and TCP traffic (Figure 121).

Analog Devices Confidential for NDA Customers Only
ADI Confidential for NDA Only
Maxim Integrated Company Confidential Under NDA
Provided to GMSL2 Customers Under NDA

Service Configuration Summary							
	Service Name	Direction	Frame Loss Rate	Max Jitter (ms)	Max Latency (ms)	Max Throughput (Mbit/s)	Service Verdict
1	Service 1	L->R	0.0E-00	0.0011		49.9996	PASS
		R->L	0.0E-00	0.0019	0.0509	49.9996	PASS
2	Service 2	L->R	0.0E-00	0.001		49.9996	PASS
		R->L	0.0E-00	0.0019	0.051	49.9996	PASS

Service Performance Summary							
	Service Name	Direction	Frame Loss Rate	Max Jitter (ms)	Max Latency (ms)	Avg Throughput (Mbit/s)	Service Verdict
1	Service 1	L->R	0.0E-00	< 0.001		49.9995	PASS
		R->L	0.0E-00	0.0011	0.0511	49.9996	PASS
2	Service 2	L->R	0.0E-00	< 0.001		49.9996	PASS
		R->L	0.0E-00	0.001	0.0511	49.9996	PASS

Figure 121. EtherSAM Service Configuration and Service Performance Tests Results

The Sub-Test section of the reports show some details about ramp and burst data (Figure 122).

Sub-Tests			
Service Configuration Test		Service Performance Test	
Ramp Test	Enabled	Status	Enabled
Burst selection	Enabled	Duration	00d:00:10:00
Duration (s)	35		

Ramp	
Step Time (s)	5
Step Values	
1	10 %
2	25 %
3	50 %
4	75 %
5	90 %
6	100 %

Burst	
Number of Burst Sequence	2
Refill Delay Ratio (%)	50
Burst/IR Frame Ratio (%)	90

Figure 122. EtherSAM Sub-Tests Results

ADI Confidential for New vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_ Customers Under NDA

The Services Summary shows the Frame Size Max Jitter, Max Latency, and Frame Loss (Figure 123).

Services Summary									
	Service Name	Profile	Frame Size (Bytes)	Direction	CIR (Mbit/s)	Max Jitter (ms)	Max Latency (ms)	Frame Loss Rate	CBS (Bytes)
1	Service 1	Data	EMIX	L->R	50.0000	2.0		1.0E-03	12144
				R->L	50.0000	2.0	15.0	1.0E-03	12144
2	Service 2	Data	EMIX	L->R	50.0000	2.0		1.0E-03	12144
				R->L	50.0000	2.0	15.0	1.0E-03	12144

Figure 123. EtherSAM Services Summary

Additional details are available in the EtherSAM reports.

29.4 Feature Availability by Device Family

RMII and RGMII are supported by select GMSL2 devices (Table 145).

Table 145. RMII/RGMII Support by Part Number

Product Family	RMII/RGMII
HDMI Serializers	
Advanced HDMI Serializers	MAX96759 MAX96761
DSI Serializers	
Advanced DSI Serializers	
CSI-2 Serializers	
Advanced CSI-2 Serializers	
CSI-2 Camera Deserializers	MAX96784
CSI-2 Display Deserializers	
OLDI Deserializers	
eDP/DP Deserializers	MAX96776 MAX96778
CSI-2 Quad Deserializers	

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

Bandwidth Calculations

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30 GMSL2 Link System Bandwidth

30.1 Overview

GMSL2 systems provide robust serial link connections between camera and display interfaces. Each interface in these interconnected systems (i.e., input interface, GMSL2 link, and output interface) has its own bandwidth requirements. Ensuring proper operation of the system requires that all bandwidth consumption be compatible and within specified limits. The overall system bandwidth is limited by the most restrictive interface: this can be either the serializer input interface, deserializer output interface, or the serial link itself.

The input and output interfaces have protocol-specific bandwidth requirements. In order to ensure compatibility with GMSL2 systems, these requirements and figures must be translated into values used by GMSL2 systems. The most important factor here is the pixel clock (PCLK). The PCLK of video received at the input interface is used to determine the video bandwidth consumed on the serial link. The GMSL video bandwidth figure is then converted into the protocol of the output interface to ensure compatibility with remote devices.

GMSL2 link bandwidth is shared by the forward channel video and other sideband channels, and both must be considered to ensure that the combined bandwidth consumption remains within protocol limits. This consideration requires evaluating two related and interactive concepts: payload size and bandwidth consumption. Payload size is the discrete size of a channel's data and is a product of the data received at the input interface; bandwidth is the total consumption by a channel and includes the payload size with the addition of encoding and other protocol overhead. Proper operation of a GMSL2 system requires that the cumulative bandwidth usage of all channels does not exceed GMSL2 protocol limits.

The following image (*Figure 124*) and table (*Table 146*) show an example of the integrated calculations necessary for determining bandwidth compliance of a serial link system.

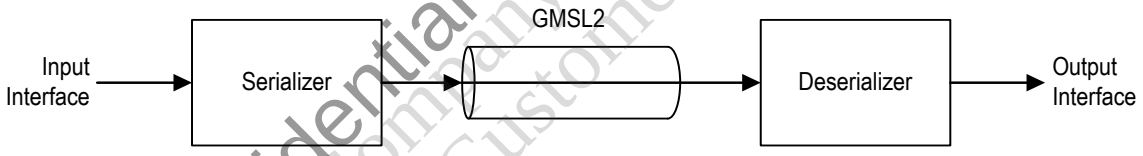


Figure 124. GMSL2 System Block Diagram

Figure 124 is a visual representation of the serial link system stages referenced in the *Table 146* example calculations used to demonstrate the concepts of payload and bandwidth throughout a serial link system. In this example, a GMSL2 system consisting of an HDMI serializer and an eDP/DP deserializer is transmitting a 1080p RGB888 video with I2C control channel information and a 1Mbps switching GPIO.

Analog Devices Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA

Table 146. Example Calculations for 1080p, RGB888 video for an HDMI Serializer and an eDP/DP Deserializer

	Input Interface (HDMI)	GMSL link	Output Interface (eDP)
PCLK	148.5MHz	148.5MHz	148.5MHz
Video Payload (PCLK * bpp)	3.56Gbps	3.56Gbps	3.56Gbps
Video Bandwidth Consumed (video payload + interface overhead)	4.45Gbps	4.04Gbps	4.45Gbps
Total Bandwidth Consumed (video bandwidth + additional interface bandwidth)	4.45Gbps	4.084Gbps*	4.45Gbps

* GMSL Link Total Bandwidth Consumed Calculation:

$$4.04Gbps + I2C + GPIO = Total\ GMSL\ Link\ Bandwidth$$

$$4.04Gbps + 4Mbps + 40Mbps = 4.084Gbps$$

See *GMSL2 Link Bandwidth Consumption from Side Channels* for additional information.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.2 GMSL2 Link Bandwidth Consumption from Video

The majority of GMSL2 serial link bandwidth comprises video transmission. The total link bandwidth consumed by video is derived from the incoming video stream and calculated by multiplying the pixel clock (PCLK) expressed in MHz, bits per pixel (bpp), and GMSL2 link overhead factors. Note that optional features (e.g., forward error correction and display stream compression) affect link bandwidth consumption and must be considered if enabled. For details on PCLK calculation, refer to the Video Basics equations section.

Note: Video transmission consumes the largest proportion of serial link bandwidth; however, all data transmitted on the serial link must be considered when calculating link bandwidth to ensure optimum serial link performance.

The equation used to calculate the forward channel video payload is:

$$Video\ Payload = PCLK * bpp$$

The equation used to calculate the total forward channel video bandwidth is shown below:

$$Video\ BW = [(video\ payload) + (video\ packet\ header) + (video\ pixel\ CRC)] * (9b10b\ encoding) * (sync\ words) * [(FEC) * (DSC)]$$

$$Video\ BW = \text{PCLK} * \left[(bpp) + \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right)\right] * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) * \left[\left(\frac{128}{120}\right) * \left(\frac{1}{3}\right)\right]$$

$$Video\ BW = \text{PCLK} * \left[(bpp) + \frac{1}{2} + \frac{1}{2}\right] * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) * \left[\left(\frac{128}{120}\right) * \left(\frac{1}{3}\right)\right]$$

Note: Video Pixel CRC, FEC, and DSC are optional features; reference *Forward Error Correction (FEC)* and *Display Stream Compression (DSC)* sections for additional information, and reference device-specific datasheets for availability details. Video **pixel** CRC is disabled by default as video **line** CRC is enabled by default and is typically preferred. Reference the Video Data CRC section for additional details.

Note: BPP = 9 is the minimum allowed bpp value for GMSL2 link BW calculation. For video datatypes with a bpp value less than 9 (e.g., RAW8 and EMB8), a bpp value of 9 must be used when calculating GMSL2 link bandwidth consumption.

This forward channel video bandwidth calculation is applicable to all input interfaces.

Forward channel rates vary depending on GMSL2 mode. *Table 147* lists the maximum video payload for each mode. The maximum rates factor in sufficient margin to accommodate sideband channels.

Table 147. Maximum Video Payload for GMSL2 Modes

GMSL2 Mode	Maximum Video Payload
3Gbps Mode	2.6Gbps (2600Mbps)
6Gbps Mode	5.2Gbps (5200Mbps)
Dual Link 6Gbps Mode	10.4Gbps (10,400Mbps)

ADI Confidential for New Vision under NDA
Analog Devices Confidential and Proprietary Information. For Use Only Under NDA
Maxim Integrated Confidential and Proprietary Information. For Use Only Under NDA
Provided to GMSL2 Customers Under NDA

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

A summary of maximum PLCK values given different bandwidth-limiting factors is presented in *Table 148*. The HDMI version is listed for applicable devices and can be ignored if the device does not support HDMI.

Table 148. Bandwidth Limiting Factors

PCLK (MHz)	HDMI Version	Video payload*	DSC** Enabled	GMSL Link Bandwidth Consumed	Minimum GMSL Link Mode Required	Limiting Interface
95MHz	1.4	2.28Gbps	No	2.59Gbps	3Gbps	3Gbps GMSL2 link
104MHz	1.4	2.50Gbps	No	2.83Gbps	6Gbps	Maximum PCLK for "H" devices
148.5MHz	1.4	3.56Gbps	No	4.04Gbps	6Gbps	VESA standard 1080p
160MHz	1.4	3.84Gbps	No	4.36Gbps	6Gbps	Maximum PCLK for "F" or "G" devices
215MHz	1.4	5.16Gbps	No	5.914Gbps	6Gbps	6Gbps GMSL2 link
340MHz	1.4	8.16Gbps	No	9.26Gbps	Dual link 6Gbps	HDMI 1.4 maximum PCLK***
340MHz	1.4	8.16Gbps	Yes	3.29Gbps	6Gbps	HDMI 1.4 maximum PCLK
430MHz	2.0	10.32Gbps	No	11.83Gbps	Dual link 6Gbps	Maximum uncompressed dual GMSL 6G link
430MHz	2.0	10.32Gbps	Yes	3.94Gbps	6Gbps	N/A (Note: Compare to video payload without DSC enabled.)
594MHz	2.0	14.3Gbps	Yes	5.75Gbps	6Gbps	VESA standard 4K
600MHz	2.0	14.4Gbps	Yes	5.81Gbps	6Gbps	Maximum PCLK supported by GMSL2; Maximum HDMI 2.0 PCLK.

* Video payload = PCLK * bpp (Note: bpp = 24 for RGB888 video)
** Display Stream Compression (DSC) provides a 3:1, visually-lossless compression of the video stream transmitted over the serial link.
*** An HDMI serializer operating in HDMI 1.4 mode has a maximum PCLK value of 340MHz. In order to transmit this video without compression, dual link mode must be used. Here, the PCLK is halved and the video is transmitted across two GMSL2 links (PCLK = 170MHz/link). This accommodates FHD (1920x1080p60, PCLK=148.5MHz) and WUXGA (1920x1200p60, PCLK=158MHz), but not 3K (2560x1440p60, PCLK=260MHz).

ADI Confidential for Review under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

30.3 Interface-Specific Bandwidth Calculations

30.3.1 HDMI Serializer Input Bandwidth

GMSL2 HDMI serializers receive TMDS (Transition-Minimized Differential Signaling) video input from HDMI sources via the HDMI input block. TMDS is an 8b/10b encoding scheme that utilizes differential signaling. This section describes the calculations necessary to convert HDMI bandwidth figures to GMSL data rates.

GMSL2 HDMI serializers have a limitation on the maximum pixel clock rate allowed. Therefore, to determine if a given video resolution is possible, the PCLK should be calculated and compared to the maximum clock rate allowed in that device.

Note: Reference the *HDMI Serializers* section for more information.

30.3.1.1 Pixel Clock Calculation

The pixel clock for GMSL2 HDMI serializers can be calculated by multiplying the horizontal and vertical resolutions (with blanking) by the frame rate (see the Video Basics section for more information):

$$PCLK = H_{total} * V_{total} * frame\ rate$$

30.3.1.2 Maximum Pixel Clock Rate (HDMI Versions)

The HDMI specifications define a maximum PCLK that can be used for each version.

- HDMI 1.0/1.1/1.2 165MHz
- HDMI 1.3/1.4 340MHz
- HDMI 2.0/2.1 600MHz

GMSL2 devices support HDMI 1.4 and HDMI 2.0. In HDMI 1.4 mode, three differential data channels operate at 10x the pixel clock frequency (up to 3.4Gbps), and the TMDS clock signal pair has the same frequency as the pixel clock. In HDMI 2.0 mode, the frequency of the TMDS clock signal pair is divided by four, and the data channels have a rate that is 40x the TMDS clock rate.

30.3.1.2.1 HDMI 1.4

- Maximum data rate (bandwidth): 10.2Gbps encoded (3.4Gbps per differential lane)
- Raw data rate (video payload): 8.16Gbps
- Maximum PCLK (24bpp): 340MHz

30.3.1.2.2 HDMI 2.0

- Maximum data rate (bandwidth): 18.0 Gbps encoded (6.0Gbps per differential lane)
- Raw data rate (video payload): 14.4Gbps raw data
- Maximum PCLK (24bpp): 600MHz

30.3.1.3 HDMI Clock Calculation

- HDMI 1.4 and below: the TMDS clock (HDMI clock) rate equals the pixel clock rate.
- HDMI 2.0: the TMDS clock (HDMI clock) rate is ¼ the pixel clock rate.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Confidential,
Provided to GMSL2 Customers Under NDA

30.3.1.4 Bandwidth Limiting Factors for HDMI Serializers

Video payload and bandwidth calculations for a 215MHz PCLK video at 24-bit color (RGB888):

Video payload:

$$215MHz * (24\ bpp) = 5160Mbps$$

Video bandwidth:

$$(5160Mbps + 107.5Mbps) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 5856Mbps\ (FEC\ disabled, No\ DSC)$$

$$(5160Mbps + 107.5Mbps) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) * \left(\frac{128}{120}\right) = 6246Mbps\ (FEC\ enabled, No\ DSC)$$

Therefore, the maximum video payload for GMSL2 HDMI display devices without FEC is:

$$5.2Gbps\ (5200Mbps)\ for\ GMSL2\ 6Gbps\ Mode$$

Note: The maximum video payload of 5.2Gbps provides a small guard band to ensure optimum performance of the serial link.

30.3.1.5 Example: HDMI Serializers

A VESA-standard full HD video stream (1080p) has a resolution of 1920x1080 at 60fps. This resolution has a total horizontal timing of 2200 pixels (i.e., 1920 active pixels and a 280-pixel blanking interval) and a total vertical timing of 1125 lines (i.e., 1080 active lines and a 45-line blanking interval).

The pixel clock is calculated by multiplying the timing by the frame rate (60fps):

$$horizontal\ timing\ * vertical\ timing * frame\ rate = PCLK(MHz)$$

$$2200 * 1125 * 60 = 148.5MHz$$

The video payload is calculated by multiplying the pixel clock by the video stream color formatting. GMSL2 display devices support 24-bit color (RGB888). The video payload is:

$$148.5MHz * (24\ bpp) = 3564Mbps$$

The overhead factors are factored in to calculate the total video bandwidth:

$$(3564Mbps + 74.25Mbps) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 4045Mbps\ (FEC\ disabled, No\ DSC)$$

$$(3564Mbps + 74.25Mbps) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) * \left(\frac{128}{120}\right) = 4315Mbps\ (FEC\ enabled, No\ DSC)$$

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

30.3.2 OLDI Deserializers Bandwidth

In GMSL2 dual OLDI deserializers, the OLDI transmit block clock rate is limited to 150MHz per OLDI port. For a RGB888 video, the OLDI clock frequency is equal to the pixel clock frequency. Therefore, GMSL2 dual OLDI deserializers are limited to a 150MHz PCLK in single OLDI mode and a 300MHz PCLK in dual OLDI mode. The minimum PCLK per port is 6.25MHz.

The VID_LOCK register on the deserializer indicates that a valid video stream is received and the OLDI output is locked to the input stream.

30.3.2.1 Example: OLDI Deserializers

Most automotive panels have a maximum OLDI clock rate of 75MHz to stay below the FM frequency band for EMI/EMC considerations. It is best practice to stay below this threshold to ensure compatibility. For example, a 1080p60 display (1920x1080x60Hz) with 1.2 blanking corresponds to a PCLK of 148.5MHz. Although this bandwidth can fit on a single OLDI output port of a GMSL2 deserializer, most 1080p panels would require a dual OLDI input with a 74.25MHz OLDI clock frequency to remain within the 75MHz limit.

30.3.3 eDP/DP Deserializers Bandwidth

The four available lanes on the eDP/DP output port are each capable of speeds of up to 8.1Gbps, resulting in a total possible bandwidth of 32.4Gbps. Subtracting the 8b10b eDP/DP protocol overhead, a maximum of 25.9Gbps of video payload is available. This figure covers all possible GMSL2 use cases (even when using compression). Therefore, the eDP/DP deserializer output port should never be a bandwidth limiting factor if all four lanes are used. If fewer lanes are enabled, the bandwidth should be calculated.

If less than the four available lanes are used, the eDP/DP deserializer output bandwidth must be calculated to ensure that the output configuration can accommodate the video stream received from the GMSL link. The data rate of the eDP/DP output port is independent of the video stream PCLK received from the GMSL link; however, it must be configured in conjunction with the number of active lanes to ensure that the deserializer can output the eDP/DP stream with 8b10b encoding. Use the eDP/DP video payload to calculate the lane rate and the number of lanes:

$$DP\ Video\ Payload = \frac{Lane\ rate * Number\ of\ lanes}{eDP/DP\ protocol\ overhead}$$

$$DP\ Video\ Bandwidth = Lane\ rate * Number\ of\ lanes$$

30.3.3.1 Example: eDP/DP Deserializers

The maximum PCLK supported in eDP/DP deserializers is 600MHz, which corresponds to a video payload of 14.4Gbps. If the DP interface is configured to four lanes at 5.4Gbps per lane, the max video payload output on the DP link is:

$$DP\ Video\ Payload = \frac{(5.4Gbps) * (4\ lanes)}{\left(\frac{10}{8}\right)} = 17.3Gbps$$

Therefore, the eDP/DP interface should not be the bandwidth-limiting factor in a system. Note that the DP sink device lane rate support must also be considered.

ADI Confidential for New Version under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA

30.3.4 DSI and CSI-2 Bandwidth Calculations

The pixel clock for MIPI input interfaces is determined by the input MIPI speed (i.e., the MIPI clock multiplied by the number of MIPI D-PHY or C-PHY data lanes). For a given video format, higher MIPI speeds allow margin for the image sensor to transmit the required video stream; however, this results in a higher PCLK value and increased consumption of GMSL2 link bandwidth. This balance means that careful consideration is required, especially for marginal system design.

30.3.4.1 DSI and CSI-2 Serializers – D-PHY Input

This section explains how the received MIPI video stream relates to the resulting PCLK frequency and bpp that feeds into the GMSL2 transmitter block. Here, the input data stream rates and packet formats are converted to GMSL formatting.

The maximum allowed PCLK is 600MHz. The MIPI D-PHY can use up to four available lanes; each lane has a maximum bandwidth of 2.5Gbps.

The PCLK is calculated with the following equation:

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

The total MIPI data rate is the product of the video PCLK and bpp:

$$Total\ MIPI\ data\ rate = LANE_CNT * LANE_RATE = PCLK * bpp$$

The maximum total MIPI data rate supported on GMSL devices is:

$$Total\ MIPI\ data\ rate = LANE_CNT * LANE_RATE = 4 * 2.5Gbps = 10Gbps$$

Note: A higher MIPI lane rate results in a higher PCLK, which consumes more bandwidth on the GMSL link. When configuring the MIPI lane rate in the video source, use the slowest lane rate possible which supports the video bandwidth (with sufficient margin).

The bpp is the bits per pixel of the chosen CSI-2 or DSI datatype. Example bpp values for various datatypes are given below. Note that, for purposes of bandwidth calculations, DSI can be considered equal to the RGB888 CSI-2 data stream (bpp = 24).

- RGB888: 24 bpp
- RAW12: 12 bpp
- RAW8: 8 bpp
- EMB8: 8 bpp

ADI Confidential for New vision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

30.4 The CSI-2 receiver recreates the pixels from the bytes in the D-PHY long packets according to the datatype's bpp value. Since each datatype can have a different bpp value, each long packet can have a different bpp value. If a MIPI video input stream consists of multiple datatypes, the long packets will have different PLCK frequencies according to the associated bpp values. GMSL systems have several modes to manage streams with heterogenous PLCK values. Note that this section focuses on the impact these modes have on bandwidth; the detailed configuration of these modes is explained in the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

30.4.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable

Analog Devices Confidential Information
Maxim Integrated Confidential,
Provided to GMSL2_Customers Under NDA
ADI Confidential for Review Only Under NDA

selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)

Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Vision under NDA

0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock, lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

30.4.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA

0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for New Version Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

30.5 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 section.

30.5.1.1.1 Managing Multiple PCLK Values in a MIPI Input Stream (Time Multiplexed)

In the following modes, the video stream is time multiplexed: the data is sequential as it would typically be transmitted from a single camera source.

30.5.1.1.1.1 Constant BPP Video Pipe Mode

This is the default mode for processing streams containing multiple PCLK values. In this basic mode, every video pipe is arranged to have a single bpp value. The maximum allowed PCLK is 600MHz regardless of the GMSL2 bandwidth. The resulting maximum allowed MIPI rates for the given D-PHY input is provided in Table 149.

$$PCLK = \frac{Total\ MIPI\ Data\ Rate}{bpp}$$

Table 149. Maximum CSI-2 Lane Rates (Mbps) in Constant BPP Mode

Max CSI-2 LANE RATE		Number of Lanes			
		1	2	3	4
bpp	8	2500.00	2400.00	1600.00	1200.00
	10	2500.00	2500.00	2000.00	1500.00
	12	2500.00	2500.00	2400.00	1800.00
	14	2500.00	2500.00	2500.00	2100.00
	16	2500.00	2500.00	2500.00	2400.00
	18	2500.00	2500.00	2500.00	2500.00
	20	2500.00	2500.00	2500.00	2500.00
	24	2500.00	2500.00	2500.00	2500.00

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Limited Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.5.1.1.1.2 Double Loading Mode

Double loading mode provides increased bandwidth efficiency and expanded use case possibilities. This data arrangement mode is available for low-bpp video datatypes with a bpp value of 8, 10, or 12. Double mode concatenates two input pixels so that they are processed as a single pixel in a video pipe. This concatenation enables video streams with heterogenous PCLK values to be combined into a single video pipe. Double loading mode also provides serial link bandwidth efficiency gains for single video streams.

Note: Double loading mode is one method of combining two video streams in a single video pipe. This allows multiple datatypes differing by a factor of two to be sent through a single video pipe without loss in bandwidth efficiency.

Example: RAW12 and RGB888 Double Loading

An image sensor transmits RAW12 (bpp = 12) and RGB888 (bpp = 24) video packets over two lanes, each operating at 1Gbps. The RAW12 packets are double loaded so that the bpp of the RAW12 packets matches the bpp of the RGB888 packets. The two datatypes are transmitted through the same video pipe.

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{2 * 1Gbps}{24\ bpp} = 83.33MHz\ (note: PCLK_{max} \leq 600MHz)$$

The link bandwidth calculation uses the highest bpp value of the transmitted datatypes. Here, the RGB888 bpp value of 24 is used.

$$BW = 83.33MHz * (24\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 2.27Gbps$$

The GMSL2 link bandwidth consumption of RAW12 and RGB888 datatypes is 2.27Gbps.

ADI Confidential for Revision Under NDA
Analog Devices Confidential, Distribution Unauthorized
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Example: RAW12 Double Loading

An image sensor transmits RAW12 (bpp = 12) video packets over one lane at 1.5Gbps.

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{1 * 1.5Gbps}{12\ bpp} = 125MHz\ (note: PCLK_{max} \leq 600MHz)$$

GMSL2 bandwidth consumption is calculated with the following equation:

$$BW = 125MHz * (12\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 1.737Gbps$$

However, the RAW12 packets can be double loaded to reduce the link bandwidth consumption (by reducing protocol overhead).

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{1 * 1.5Gbps}{24\ bpp} = 62.5MHz\ (note: PCLK_{max} \leq 600MHz)$$

GMSL2 bandwidth consumption is calculated with the following equation:

$$BW = 62.5MHz * (24\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 1.702Gbps$$

In this example, double loading the RAW12 packets provides a bandwidth reduction of 35Mbps.

ADI Confidential for New vision under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.5.1.1.1.3 Zero-Padding Mode

Zero-padding allows multiple datatypes that do not have the same bpp value to be transmitted on a video pipe together. This option is an alternative to double loading mode and does not require the bpp values to differ by a factor of two. With zero-padding, the datatype with the lower bpp value is zero-padded to be the same length as the bpp of the larger datatype.

The PCLK is derived from the datatype with the lowest bpp value. Prior to zero-padding one of the datatypes, there are two PCLKs due to the two different bpp values. The slower video stream (i.e., the datatype with the larger bpp value) can only be processed with the clock from the faster video stream (i.e., the datatype with the smaller bpp value).

The link bandwidth calculation uses the bpp value of the video pipe. This is equal to the highest bpp value of the transmitted datatypes.

The video streams are time multiplexed. The multiple datatypes are not transmitted simultaneously; individual frames of each datatype are transmitted sequentially.

Note: Zero-padding mode is available for all datatypes with bpp ≤16.

Example: EMB8 and RAW12 Zero-Padding

An image sensor transmits EMB8 (bpp = 8) and RAW12 (bpp = 12) video packets over two lanes, each operating at 1Gbps. The EMB8 packets are zero-padded with four zeros per pixel, matching the bpp of RAW12. The two datatypes are transmitted through the same video pipe.

EMB8 has a bpp = 8 and RAW12 has a bpp = 12. Since the PCLK is calculated using the datatype with the lowest bpp value, the EMB8 value is used in the equation.

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{2 * 1Gbps}{8\ bpp} = 250MHz\ (note: PCLK_{max} \leq 600MHz)$$

The 250MHz value is confirmed to be smaller than 600MHz (the maximum PCLK value).

The link bandwidth calculation uses the highest bpp value of the transmitted datatypes. Here, the RAW12 bpp value of 12 is used.

$$BW = 250MHz * (12\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 3.47Gbps$$

The GMSL2 link bandwidth consumption of EMD8 and RAW12 datatypes is 3.47Gbps.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential,
Provided to GMSL2_ Customers Under NDA

30.5.1.1.1.4 Double Loading Mode in Combination with Zero-Padding Mode

Double loading mode and zero-padding mode can be used together to accommodate a wide range of use cases involving many different datatypes.

Example: An image sensor transmits RAW12 and EMB8 video packets in a two-lane CSI-2 stream, each lane operating at 1Gbps. In the serializer, The EMB8 and RAW12 packets are transmitted together in a video pipe. Here, double loading mode is used to pack two EMB8 pixels together into a 16 bpp pixel, and zero-padding is used to increase the bpp of the RAW12 packet from 12 to 16. With the effective bpp of each datatype equal to 16, both datatypes can be sent together via the same video pipe.

To calculate the PCLK, the bpp from the datatype with the lowest bpp value (i.e., higher PCLK value) is used. In this example, the RAW12 bpp value is the lowest (bpp = 12) and is used for the PCLK calculation.

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{2 * 1Gbps}{12\ bpp} = 166.67MHz$$

The link bandwidth calculation uses the highest bpp value of the transmitted datatypes. Here, the double loaded EMB8 bpp value of 16 is used.

$$BW = 166.67MHz * (16\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 3.057Gbps$$

The GMSL2 link bandwidth consumption of EMB8 and RAW12 datatypes is 3.057Gbps.

ADI Confidential for New Revision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.5.1.1.2 Managing Video that is not Time Multiplexed

The above calculations assume that the video stream is time multiplexed. If the video streams are not time multiplexed, the MIPI long packets may overlap in time. This may occur if multiple independent MIPI inputs are received by the serializer (Figure 125). To calculate the peak bandwidth consumed on the serial link, the peak bandwidth demands of each video stream are calculated separately then summed together. This figure must be less than the maximum video bandwidth limit to ensure the proper operation of the serial link.

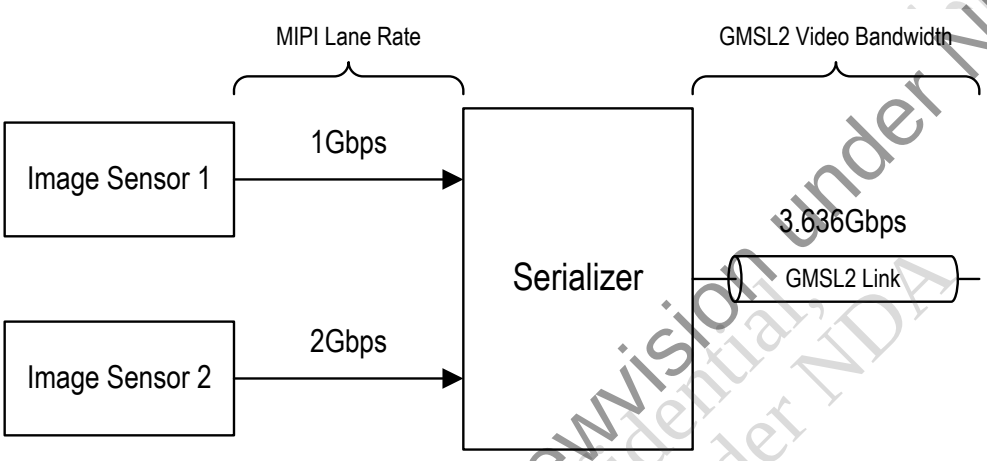


Figure 125. A GMSL2 Serializer with Two Independent MIPI Inputs

Example: Two image sensors separately transmit EMB8 (bpp = 8) and RAW12 (bpp = 12) to a single CSI-2 serializer. The EMB8 video stream has a lane rate of 1Gbps; the RAW12 video stream has a lane rate of 2Gbps.

EMB8 video stream:

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{1 * 1Gbps}{8\ bpp} = 125MHz$$

$$BW = 125MHz * (9\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 1.320Gbps$$

Note: BPP = 9 is the minimum allowed bpp value for GMSL2 link BW calculation.

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

RAW12 video stream:

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

$$PCLK = \frac{1 * 2Gbps}{12\ bpp} = 166.67MHz$$

$$BW = 166.67MHz * (12\ bpp + 0.5) * \left(\frac{10}{9}\right) * \left(\frac{2048}{2047}\right) = 2.316Gbps$$

Combined GMSL2 bandwidth consumption:

$$BW_{peak} = 1.320Gbps + 2.316Gbps = 3.636Gbps$$

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

30.5.1.2 CSI-2 Deserializers – D-PHY Output

30.6 The MIPI CSI-2 output clock frequency is independently configured. The phy_csi_tx_dpll_predef_freq registers allow adjustments with 100Mbps step resolution. The CSI2_LANE_CNT register programs the LANE_CNT definition (one to four lanes). Further configuration of the of the output frequency is covered in the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

30.6.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

ADI Confidential for New Vision Under NDA
Analog Confidential Distribute Only Under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock,

ADI Confidential Information Newvision under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customer Under NDA

					lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

30.6.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential Company Confidential Provided to GMSL2_Customers Under NDA

		watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for Newvision Under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.7 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 MIPI Output Data Rate section.

Relevant equations:

- $Total\ MIPI\ data\ rate = LANE_RATE * LANE_CNT$
- $Total\ MIPI\ data\ rate \geq PCLK * (BPP_{single\ video\ receiver\ pipe})$

If there are multiple video pipes aggregated to a single CSI-2 output, the calculation must consider if the packets are time domain multiplexed between the video pipes.

- If multiple video pipes have time-overlapping data:

$$(PCLK1 * BPP1) + (PCLK2 * BPP2) + \cdots \leq Total\ MIPI\ data\ rate$$

- If multiple video pipes have time-separated data:

$$MAX\{(PCLK1 * BPP1), (PCLK2 * BPP2), \ldots\} \leq Total\ MIPI\ data\ rate$$

The maximum allowed PCLK is 600MHz. In standard operation, the maximum MIPI Lane Rate is 2500Mbps. GMSL2 dual-link mode increases the serial link bandwidth between the serializer and the deserializer to 12Gbps. This increased bandwidth can transmit CSI-2 data to the deserializer at 10.4Gbps. Enabling DSC can further increase the CSI-2 data available to the deserializer, refer to the Display Stream Compression (DSC) section for additional information.

ADI Confidential for New Version under NDA
Analog Devices Confidential – Distribution only under NDA
Maxim Integrated Company Confidential – Under NDA
Provided to GMSL2_Customers

30.7.1.1 CSI-2 Deserializers – C-PHY Output

CSI-2 C-PHY output supports higher output bandwidth than D-PHY. C-PHY uses three-phase symbol encoding that delivers 2.28 bits per symbol over three-wire trios, each with an embedded clock (note: 'trios' are also referred to as 'lanes' or 'pins'). Each trio operates at 2.5GSym/s, resulting in a bandwidth of 5.7Gbps per trio. Lanes can be combined to increase available bandwidth (Table 150). There is no line coding overhead (e.g., 8b10b). Video pipes Y and Z permit 1x3 and 1x4 configurations, while video pipes X and U only allow 1x1 and 1x2 lane configurations (Figure 126).

Table 150. MIPI D-PHY and C-PHY Lane Configurations and Output Bandwidth

	D-PHY		C-PHY	
Width (CLKxDATA)	Pins	Rate (Gbps)	Pins	Rate (Gbps)
1x1	4	2.5	3	5.7
1x2	6	5.0	6	11.4
1x3	8	7.5	9	17.1
1x4	10	10.0	12	22.8

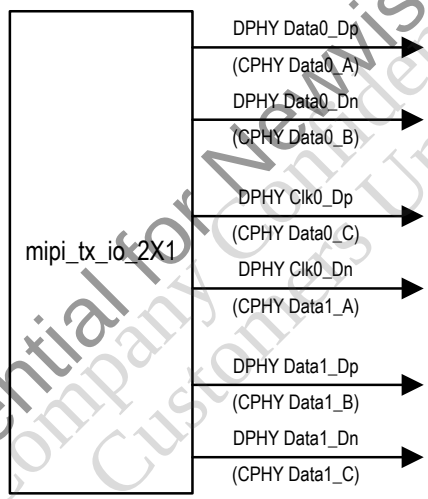


Figure 126. MIPI Lane Configurations

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distributed Under NDA
Maxim Integrated Company Confidential, Customers Under NDA
Provided to GMSL2

30.8 Pixel Clock Detection and Calculations

30.8.1 Maximum Pixel Clock Frequency by Device

- 600MHz – All RGMII/DSC HDMI serializer parts.
  - (Note: The maximum PCLK for HDMI 1.4 is 340Mhz; the maximum PCLK for HDMI 2.0 is 600Mhz)
- 450MHz – All non-RGMII/DSC HDMI serializer parts.
- 160MHz – Part numbers ending with the letter 'F' or 'R'.
- 104MHz – Part numbers ending with the letter 'H'.

Note: The above list is provided as a reference. Verify the maximum PCLK value for a given part number by confirming with the device-specific datasheet. Maximum PCLK indicated may be affected other bandwidth limitations and/or side channel consumption.

30.8.2 Pixel Clock Detection

The bitfield PCLKDET (register VIDEO_TX2) indicates if a video input clock is detected. If this field is zero, perform the following:

- Check that the video system is enabled with the VID_TX_EN registers.
- Ensure that the VIDEO TX block is correctly connected.
- Verify that the video input is operational.

GMSL2 devices detect the input PCLK frequency. Access the SPEED_DET [1:0] status register to view the range of the detected input PCLK value (Table 151).

Table 151. PCLK Value Detection

SPEED_DET [1:0] Value	Decode
0b00	PCLK is greater than 240MHz
0b01	PCLK is in the range of 160MHz – 240MHz
0b10	PCLK is in the range of 104MHz – 160MHz
0b11	PCLK is less than 104MHz

The following formula provides a more precise estimation of the detected input frequency: Read the associated registers from the VID_TX register block and apply the formula.

ADI Confidential for New Vision under NDA
Analog Confidential Document Subject to Non-Disclosure Agreement
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

30.8.3 Pixel Clock Parameters

These parameters are reported by the PCLK PLL. This reported PCLK value is an estimate with the following tolerances:

- ±10% accurate between 50MHz – 300MHz
- ±5% accurate above 300MHz

- ODIV = VID_ODIV_L[7:0] + VID_ODIV_H*0x100)
- FBDIV = FB_DIV[5:0]
- FBDIVFRAC = FB_DIV_FRAC[5:0]
- PCLK (MHz) $= \frac{75}{ODIV} * \left(FBDIV + \frac{FBDIVFRAC}{64}\right)$
- Frame rate $= \frac{PCLK * 10^6}{total\ pixels}$ *where* total pixels *is the number of pixels in a frame* $(H_{tot} * V_{tot})$

30.9 Status and Debug Registers

The following bandwidth status registers report bandwidth error conditions that can lead to degraded serial link performance.

- VIDEO_TX2: OVERFLOW – Reports video bandwidth overflow.
- VIDEO_TX2: DRIFT_ERR – Reports a drift error indicating that the PCLK frequency is not stable.

ADI Confidential for Review under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.10 GMSL2 Link Bandwidth Consumption from Side Channels

Side channels (e.g., RGMII, SPI, GPIO, UART, I2C, Audio, etc.) occupy channel bandwidth when enabled. The total link bandwidth usage is the sum of the side channel consumption, the utilization from video, and protocol overhead. This calculation must be below the specified bandwidth for the serial link to avoid overflow.

Note: Video signals only occupy the forward channel (serializer to deserializer) of the GMSL2 link, while side channels may operate on both the forward and reverse channels. Typically, the side channels consume bandwidth in the direction the data is sent – referred to here as "forward bandwidth" and also in the opposite direction – referred to here as "reverse bandwidth" if CRC is enabled.

The calculations are agnostic of GMSL link direction. For example, if audio is sent from the deserializer to the serializer, the audio forward bandwidth is consumed on the GMSL reverse channel, and the audio reverse bandwidth is consumed on the GMSL forward channel.

Note: CRC increases bandwidth consumption for all side channels (in both directions) and is enabled by default for all channels besides RGMII. Disabling CRC is not recommended for most applications. Reference the *CRC Error Detection and ARQ Error Correction* section for additional details.

30.10.1 Audio

Audio bandwidth consumption is a function of the audio sample rate and sample depth. Bandwidth is consumed in the direction of the audio channel as well as the reverse direction for the acknowledge packets.

Audio Forward BW (Mbps) = $\left(\frac{\text{sample_rate}}{1000}\right) * 20 * \left(\text{roundup}\left(\text{channels} * \frac{\text{sample_depth}}{18} + 0.5\right) + 1 + \text{CRC}\right)$

Audio Reverse BW (Mbps) = $\left(\frac{sample_rate}{1000}\right) * 6 * 20 * \text{CRC}$

Where:

- sample_rate = audio sample rate (kbps).
- sample_depth = audio sample depth (bits per sample).
- channels = number of channels.
- roundup() is a function defined as rounding up the contained number to the next integer value.
- CRC = 1 if CRC is enabled; CRC = 0 if CRC is not enabled.

30.10.1.1 Example: Audio Forward Bandwidth

An eight-channel audio signal with a 192kbps sample rate and 32-bit sample depth is transmitted in the forward direction. The following equation is used to calculate the forward bandwidth consumption:

BW (Mbps) = $\left(\frac{192}{1000}\right) * 20 * \left(roundup\left(channels * \frac{32}{18} + 0.5\right) + 1 + 1\ (CRC\ is\ enabled\ by\ default)\right) = 65.28 Mbps$

Maxim Confidential for Review under NDA
Analog Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

30.10.2 GPIO

GPIO bandwidth consumption is a function of how many transitions per second occur. This consumption is increased if delay-compensation mode is enabled. Bandwidth is consumed in the direction of the transmitted GPIO transmission; however, if CRC/ARQ is enabled, the ARQ acknowledge packets also consume reverse channel bandwidth.

GPIO Forward BW (Mbps) = $(T * (2 + COMP + CRC) * 20)/1000000$

GPIO Reverse BW (Mbps) = $(T * CRC * 2 * 20)/1000000$

Where:

- T = transitions per second.
- COMP = 1 if delay compensated mode is enabled; COMP= 0 if delay compensated mode is disabled.
- CRC = 1 if CRC is enabled; CRC = 0 if CRC is not enabled.

For example, if a GPIO is used to transmit a touch interrupt signal, the maximum transitions per second would be approximately five, resulting in a bandwidth consumption of 0.3kbps on the forward channel.

Note: For serializers the Forward BW is consumed from serializer to deserializer and Reverse BW is consumed from deserializer to serializer. For deserializers the opposite is true, the Forward BW is consumed from deserializer to serializer and Reverse BW is consumed from serializer to deserializer.

30.10.3 SPI

SPI bandwidth consumption is a function of the SPI clock and byte length. Bandwidth is consumed in the direction of the SPI transaction as well as the reverse direction for the acknowledge packets.

SPI Forward BW (Mbps) = $(fSCK/(8 * L)) * (2 + CRC + roundup((L - 1)/2)) * 20$

SPI Reverse BW (Mbps) = $(fSCK/(8 * L)) * 2 * 20 * CRC$

Where:

- fSCK = SPI clock (SCK) frequency in Mbps.
- L = SPI read/write length in bytes.
- roundup() is a function defined as rounding up the contained number to the next integer value. CRC = 1 if CRC is enabled; CRC = 0 if CRC is not enabled.

Note: For serializers the Forward BW is consumed from serializer to deserializer and Reverse BW is consumed from deserializer to serializer. For deserializers the opposite is true, the Forward BW is consumed from deserializer to serializer and Reverse BW is consumed from serializer to deserializer.

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential, Provided under NDA GMSL2_Customers Under NDA

30.10.4 RGMII

The RGMII bandwidth consumption is a function of the ethernet speed, utilization percentage, and the REQ_Ratio. Ethernet is a bi-directional interface: bandwidth is consumed in both the forward and reverse directions.

RGMII Forward BW (Mbps) = $ethernet_speed * \left(\frac{1522}{1534}\right) * max_for_utilization * REQ_Ratio_{max} * \left(\frac{9}{8}\right) * \left(\frac{10}{9}\right)$

RGMII Reverse BW (Mbps) = $ethernet_speed * \left(\frac{1522}{1534}\right) * max_rev_utilization * REQ_Ratio_{max} * \left(\frac{9}{8}\right) * \left(\frac{10}{9}\right)$

Note: If CRC is enabled it will consume additional bandwidth. Use of CRC on RGMII packets is discouraged as RGMII has error detection and retransmission mechanisms at a higher protocol level.

Where:

- ethernet_speed is the configured ethernet speed (10Mbps, 100Mbps, or 1000Mbps) in Mbps. max_utilization is the maximum percentage of the ethernet bandwidth consumed in each direction. For example, if the maximum data transmitted over the ethernet link is 50Mbps in the forward channel when ethernet_speed is configured to 100Mbps mode, the max_forward_utilization value is 0.5.
- REQ_Ratio_{max} is calculated using the REQ Hold Off value (configured via registers). Reference the RMII/RGMII (Ethernet) section for details. Calculate REQ_Ratio_{max} value from the following equations:
    - $REQ_Ratio_{max} = \max(REQ_Ratio_1, REQ_Ratio_2)$
        - $REQ_Ratio_1 = (REQ\ Hold\ Off + 2)/(REQ\ Hold\ Off + 1)$
        - $REQ_Ratio_2 = (REQ\ Hold\ Off + 4)/(REQ\ Hold\ Off + 2)$

- **Note:** For serializers the Forward BW is consumed from serializer to deserializer and Reverse BW is consumed from deserializer to serializer. For deserializers the opposite is true, the Forward BW is consumed from deserializer to serializer and Reverse BW is consumed from serializer to deserializer.

30.10.5 I2C

The I2C bandwidth consumption is a function of the SCL frequency. The I2C is generated simultaneously in both directions and includes ARQ ACK traffic; bandwidth is consumed in both the forward and reverse directions.

I2C Forward BW (Mbps) = $(fSCL/4500) * (2 + CRC) * 20$

I2C Reverse BW (Mbps) = $40 * (fSCL/1000) * CRC$

Where:

- fSCL is the I2C clock rate (kbps).
- CRC = 1 if CRC is enabled; CRC = 0 if CRC is not enabled.

Note: For serializers the Forward BW is consumed from serializer to deserializer and Reverse BW is consumed from deserializer to serializer. For deserializers the opposite is true, the Forward BW is consumed from deserializer to serializer and Reverse BW is consumed from serializer to deserializer.

ADI Confidential for NDA Version under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

30.10.6 UART

UART bandwidth consumption is a function of the UART baud rate. UART consumes bandwidth in in the direction of the UART transaction as well as the reverse direction for the acknowledge packets.

UART Forward BW (Mbps) = $(fUART/10000) * (2 + CRC) * 20$

UART Reverse BW (Mbps) = $(fUART/10000) * 2 * 20$

Where:

- fUART is the UART baud rate.
- CRC = 1 if CRC is enabled; CRC = 0 if CRC is not enabled.

Note: For serializers the Forward BW is consumed from serializer to deserializer and Reverse BW is consumed from deserializer to serializer. For deserializers the opposite is true, the Forward BW is consumed from deserializer to serializer and Reverse BW is consumed from serializer to deserializer.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribute Only to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

30.11 Dynamic Bandwidth Allocation

30.11.1 GMSL2 Channel Priority Settings

GMSL2 devices allow for the assignment of priority levels to different communications channels. Priority settings are user definable; however, in most cases (i.e., when link bandwidth utilization is less than 98%), default priority settings do not need to be changed. Priority settings are most critical if more than the available bandwidth is used, as the lowest priority channel will begin to lose data. This allows the user to select which channel will begin to lose data first by assigning it the lowest priority.

In some uses cases where link bandwidth is utilized at more than 98% continuously or for a sustained period (e.g., one video line length), then adjusting the channel priority settings may facilitate a more stable configuration.

There are four priority settings available for each channel: low, normal, high, and urgent. In general, all communication channels use priority 1 (low) as default, but the priority level can incrementally increase by 1 when the internal FIFO becomes full. Channels with higher priority are always serviced before channels with lower priority. The scheduler arbitrates between the channels that have the same priority setting based on recent link utilization.

Note: Priority settings are not available for GPIOs, but are available for most other bi-directional channels

Register settings for each channel can be used to override the default priority settings. Set the desired priority value to PRIO_VAL[1:0]. Then, set PRIO_CFG = 3 to confirm that priority setting. Note that the value programmed into PRIO_VAL[1:0] does not take effect until after PRIO_CFG = 3 has been written.

30.11.2 Priority Overwrite Example

Scenario:

1. SPI transmissions should be transmitted from deserializer to serializer (reverse channel) as fast as possible (with minimum latency).
2. Only SPI and I2C are using the reverse channel.
3. SPI bit rate is low enough so that it never uses the full reverse channel bandwidth.

Then, if SPI priority is set to 3 (SPI: PRIO_CFG = 3) and I2C priority is set to 2 (CC_PRIO_CFG = 2) in the deserializer, the SPI packets will always be transmitted before I2C packets when both channels request to transmit at the same time.

Normally, an I2C packet is generated for each I2C bit; since I2C can be slowed down by clock stretching, this kind of de-prioritization is not expected to affect I2C throughput significantly.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

30.11.3 Bandwidth Sharing Programming

Note: Most use cases do not require manual programming of bandwidth sharing ratios; manual programming is only needed if the required bandwidth is very close (over 98%) to the maximum bandwidth supported by GMSL2. Reference the Interface-Specific Bandwidth Calculations section for more information.

The bandwidth ratio can be set by programming the BW_MULT and BW_VAL registers for each transmit communication channel. BW_MULT sets the multiplier to one of 1x, 4x, or 16x. BW_VAL is multiplied by BW_MULT value to get the BW_SHARE value which indicates the allocated ratio of bandwidth with respect to 1,000, where 1,000 indicates full link bandwidth (100%). For example, if BW_VAL = 15 and BW_MULT = 4x, then bandwidth share for that channel is set to 15x4 = 60/1000 = 6%.

30.11.4 Bandwidth Allocation Example

Assume that there is a burst SPI transmission request, and the link can transmit either a video or an SPI packet. If video is assigned 90% link share and SPI is assigned 5% link share, then SPI may be allowed a maximum of 10% of the link bandwidth so that video gets its fair share of 90% and the video buffers do not overflow. If the SPI link attempts to use more than 10% of the bandwidth, it will be throttled down by the scheduler.

ADI Confidential for NewVision Under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

Functional Safety

ADI Confidential for NewVision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

31 Functional Safety Introduction

Section pending.

TK Include the cross references (xref) from other section headers.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

32 ASIL

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33 GMSL2 Error Reporting (ERRB Pin)

33.1 Overview

All GMSL2 devices have a multi-purpose open-drain error reporting and interrupt status output (ERRB). The configurable ERRB pin can output detected error status(es) and/or certain internal device events relevant to the host processor (i.e., interrupts). The ERRB function is assigned to an MFP (reference the I/O Matrix (MFP) section for an overview and device-specific datasheets for pinout information).

The ERRB pin operates in open-drain output mode with a 40kΩ internal pull-up resistor. It is an active-low signal: 1 indicates “no error and no interrupt event,” 0 indicates the occurrence of an “error and/or interrupt event.”

Note: The ERRB pin reflects the status of the ERROR bit (register 0x0013) but with inverted logic (i.e., ERROR=1 and ERRB pin low indicates an error condition).

Multiple errors and event sources can be simultaneously configured to drive the ERRB pin. Each error and event able to drive the ERRB pin has a status flag register. When several errors and/or events are configured to drive the ERRB pin, these status registers can be read to determine the source of the ERRB assertion. Each ERRB source can be individually enabled and disabled.

Functions and uses include:

- Very fast routing of on-chip diagnostics to device pin used as an interrupt for the SoC.
- ASIL-related functions route to the internal ERRB signal to be used as an interrupt.
- Remote-side forwarding of diagnostics to local ERRB signal permits all GMSL2 system devices to centralize ERRB reporting to a single pin.

33.2 Operation

Most GMSL2 device errors/statuses can be output to the ERRB pin. Independent of ERRB reporting, the individual device error statuses are reported in the local registers of the feature. Many of these errors have attendant counters that track the cumulative number of errors that have occurred. Refer to device-specific datasheets for more information.

Errors are individually configured to be output to the ERRB pin. Each error eligible for ERRB pin routing has an output enable (OEN) bit that is used to send the error status to the ERRB OR gate. The OR gate controlling the ERRB status is illustrated in *Figure 127* below. Note that concept is universal to GMSL2 devices; available errors and event statuses may vary by device.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

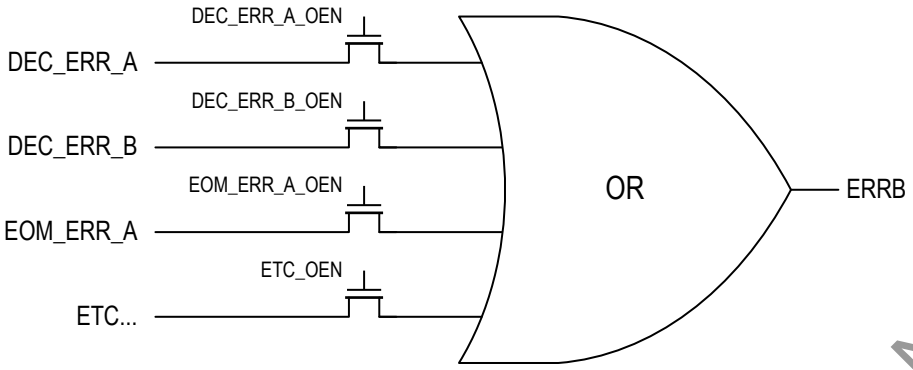


Figure 127. ERRB Logic

The ERRB pin can be used to monitor important error or status flags. If the ERRB pin is driven low and multiple errors/statuses are configured to output to ERRB, the local registers for the enabled errors/statuses should be read to determine the cause. Figure 128 shows the internal error reporting mechanism.

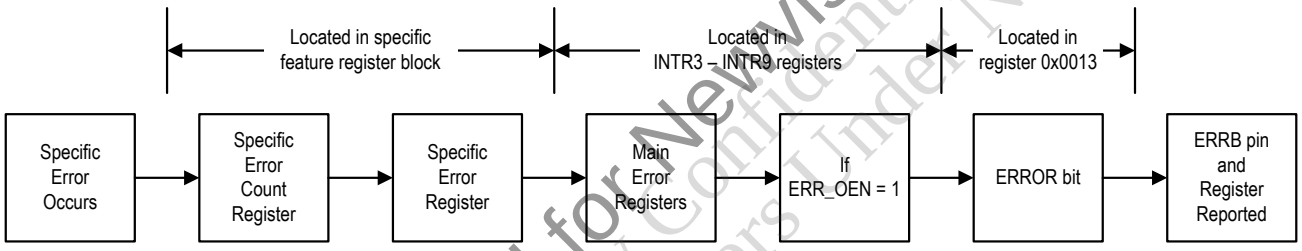


Figure 128. Internal Error Reporting Mechanism

If an error occurs and its ERR_OEN bit is enabled, the ERRB pin will be driven low (i.e., active) and the ERROR bit (register 0x0013) will report a 1. This alerts the user to a non-specific error state among the enabled error bits. To determine what error occurred and whether the error is continuous or a one-time event, a sequence of register reads of the ERRB errors/statuses must be executed. GMSL2 errors are latching: once an error occurs, an error flag is set and is not cleared until it or the associated error count register is read/cleared.

Figure 129 contains a suggested process for identifying errors resulting in an ERRB transition.

ADI Confidential for Nevision Under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

33.2.1 Example ERRB System Reaction

This example is presented to illustrate the function of errors within a GMSL2 system and describe the process used to determine the source of the error condition (Figure 129).

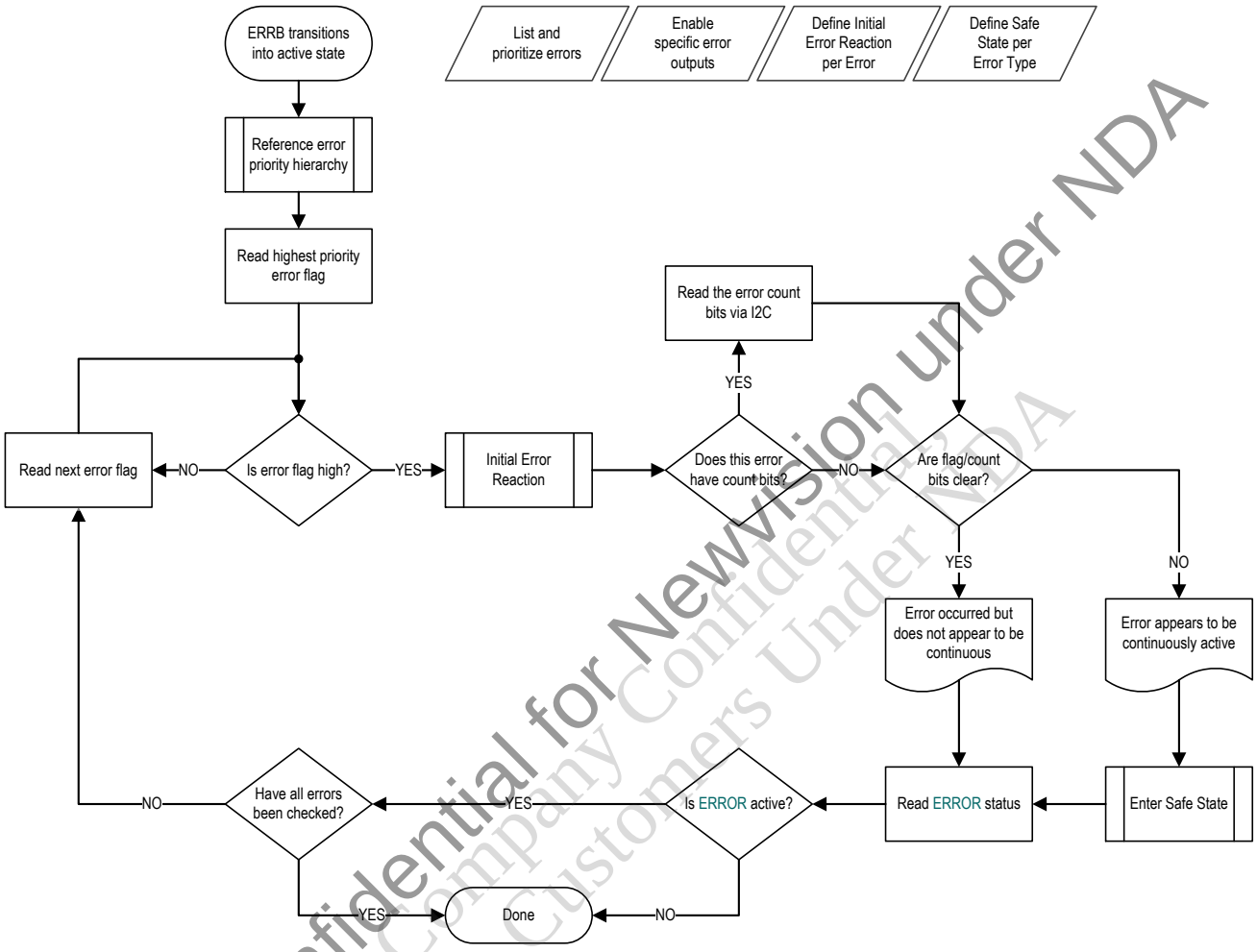


Figure 129. ERRB Transition System Reaction Process

Four monitored errors are assigned the following priority:

1. Line fault error
2. GMSL decoding error
3. EOM error
4. Watermark error

Confidential for NewVision under NDA
Maxim Integrated Company Confidential NDA
Provided to GMSL2 Customers Under NDA

During initial system configuration, enable ERRB output for the relevant errors:

- Line fault error: LFLT_INT_OEN = 1
- GMSL decoding error: DEC_ERR_OEN = 1
- EOM error: EOM_ERR_OEN = 1
- Watermark error: WM_ERR_OEN = 1
- All other error output enable (OEN) bits are set to 0.

A link quality fault event occurs and ERRB transitions to the active state. The system reaction process (defined in Figure 129) is used to determine the cause(s) of the error state.

- ERRB transitions to active state (ERROR = 1).
- The highest priority error is line fault error.
  - LFLT_ERR_FLAG is read, returns LFLT_ERR_FLAG = 0.
- The next highest priority error is decode error.
  - DEC_ERR_FLAG is read, returns DEC_ERR_FLAG = 1.
  - System reacts as defined by system engineer for decode error fault condition.
  - The decode error counter register DEC_ERR_A[7:0] is read and returns a non-zero value.
  - The decode error counter register DEC_ERR_A[7:0] is read again and returns zero, indicating this was a non-continuous error condition.
  - ERRB pin status is checked again (either by ERRB pin state or reading ERROR register value). The ERRB status is still active (ERROR = 1).
- The next highest priority error is EOM error.
  - EOM_ERR_FLAG is read, returns EOM_ERR_FLAG = 1.
  - System reacts as defined by system engineer for EOM error fault condition.
  - The EOM error does not have count bits.
  - EOM_ERR_FLAG is read again and now returns EOM_ERR_FLAG = 0, indicating this was a non-continuous error condition.
  - ERRB status is checked again and is no longer active (ERROR = 0).
- The remaining lower priority errors do not need to be checked as there is no longer an error condition present (i.e., watermark errors do not need to be checked).
- System returns to nominal operating state.

ADI Confidential for review only
Analog Devices Confidential for review only under NDA
Maxim Integrated Confidential Under NDA
Provided to GMSL2_Customers Under NDA

33.2.2 Disabling the ERRB Function

The ERRB function can be disabled if, for example, error/interrupt reporting is not required, or the MFP pin used by ERRB is needed for another function. To disable the ERRB function of the default MFP, set ERRB_EN = 0. If ERRB is disabled, the MFP pin will default to GPIO functionality. Note that the pin can be further reassigned if different functionality is required (reference the General-Purpose Input and Output (GPIO) section for more information). Select GMSL2 devices allow for the assignment of an alternate MFP to be used for ERRB output. For compatible devices, use the ALT_ERRB_EN register to enable the alternative MFP for ERRB.

33.2.3 Remote Error Reporting

After the serial link is locked, GMSL2 devices can report the ERRB status of remote device(s) via a GPIO with programmable ID. To enable remote error reporting, set ERR_TX_EN = 1, and assign the GPIO ID by programming the ERR_TX_ID register. The local device receives the transmitted error over the GPIO tunnel.

33.2.3.1 Output Error Status with Local GPIO

Configure the remote device to output the error status via GPIO tunneling. Then, program the local device with the following register writes to connect the local device to the GPIO tunnel and output the error status on an available MFP:

- Write GPIO_RX_EN = 1 to enable the GPIO receive block for the GPIO pin.
- Write GPIO_TX_EN = 0 to disable the GPIO transmit block for the GPIO pin.
- Set GPIO_OUT_DIS = 0 to enables the GPIO output driver.
- Write GPIO_TX_ID = “Remote ERR_TX_ID value” to assign the GPIO pin ID with the same GPIO pin ID as the remote-side ID transmitting the error status.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

33.2.3.2 Combine Remote and Local Error Statuses for ERRB Output

The remote and local error statuses can be combined for consolidated ERRB output on the local device. Set ERR_TX_EN = 1 in the remote device. If necessary, program the ERR_TX_ID register to an unused value. In the local device, set ERR_RX_ID = “Remote ERR_TX_ID value”, ERR_RX_EN = 1, and REM_ERR_OEM = 1.

The remote ERRB status will be transmitted to the local device and will drive the local ERRB pin active (unless already active due to a previous local error or interrupt condition). The host controller can read the REM_ERR_FLAG register to determine whether the source of the error originated in the remote or local device.

GMSL2 devices that support connections to multiple remote devices (i.e., devices with two links and serializers that support daisy chaining) can consolidate all remote ERRB status signals to the local ERRB pin. For daisy chain applications, multiple ERRB signals can be combined and output from a separate MFP (i.e., combining multiple GPIO tunnel outputs).

33.2.3.3 Auto Error Reset

Some error counters can be automatically reset after ERRB is asserted. This allows for continuous monitoring of the ERRB pin and error tracking without requiring register reads. When AUTO_ERR_REST_EN = 1, the following register counters are automatically reset after the ERRB pin is asserted for 1µs:

- DEC_ERR_A
- DEC_ERR_B
- IDLE_ERR
- VID_PXL_CRC_ERR

Similarly, when AUTO_CNT_RST_EN=1, the PKT_CNT counter register is automatically reset after the ERRB pin is asserted for 1µs. This configuration can be used to count received packets by observing the ERRB pin.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33.3 Configuration

All possible errors and events capable of driving the ERRB pin are listed below with configuration details. “Flag” indicates the name of the read-only register that is asserted to 1 in the case of an error condition; “OEN” indicates the name of the register used to enable reporting of the flag status to the ERRB pin.

Note: Although ERRB is supported by all GMSL2 devices, not all errors/interrupts addressed in this section are available in all devices. Reference device-specific datasheets for available errors and interrupts. If the register controlling error/interrupt output to the ERRB pin is not listed, that error/interrupt reporting function is not available in that device.

33.3.1 GMSL Decoding Errors

Flag:

- DEC_ERR_FLAG_A
- DEC_ERR_FLAG_B
- DEC_ERR_FLAG_C
- DEC_ERR_FLAG_D
- DEC_ERR_FLAG_SIO
- DEC_ERR_FLAG_DCIO

OEN (Output Enable):

- DEC_ERR_OEN_A
- DEC_ERR_OEN_B
- DEC_ERR_OEN_C
- DEC_ERR_OEN_D
- DEC_ERR_OEN_SIO
- DEC_ERR_OEN_DCIO

Description: Each GMSL link has a built-in 8-bit counter that tracks detected 9b10b disparity errors. Disparity errors are detected whenever the 9b10b running disparity exceeds the minimum or maximum limit.

Note: Isolated single-bit errors are detected with a high probability. Burst errors and single-bit errors occurring in close proximity may not be detected or may be counted as a single error. Therefore, decode errors cannot be used to calculate bit error rate (BER) statistics. Reference *BER Testing using the GMSL2 Idle Link* (*PRBS Testing* section) for additional information.

Decoding error counter registers:

- DEC_ERR_A
- DEC_ERR_B
- DEC_ERR_C
- DEC_ERR_D
- DEC_ERR_SIO
- DEC_ERR_DCIO

Each decoding error flag is asserted when the corresponding decoding error counter value is greater than or equal to the error counter threshold value defined by DEC_ERR_THR (default is 1).

Related registers: DEC_ERR_THR, AUTO_ERR_RST

Clearing: Decoding error counters are reset to 0 when read, reset by the Auto Error Reset function, or after establishing link lock after loss of lock. After a decoding error counter is reset, its error flag is also reset (the counter value of 0 is always less than the error threshold value).

Maxim Integrated Confidential for New Version under NDA
Maxim Integrated Company Confidential Under NDA
Provided to GMSL2_Customers Under NDA

33.3.2 GMSL Idle Packet Errors

Flag:

- IDLE_ERR_FLAG
- IDLE_ERR_FLAG_A
- IDLE_ERR_FLAG_B
- IDLE_ERR_FLAG_C
- IDLE_ERR_FLAG_D
- IDLE_ERR_FLAG_SIO
- IDLE_ERR_FLAG_DCIO

OEN (Output Enable):

- IDLE_ERR_OEN
- IDLE_ERR_OEN_A
- IDLE_ERR_OEN_B
- IDLE_ERR_OEN_C
- IDLE_ERR_OEN_D
- IDLE_ERR_OEN_SIO
- IDLE_ERR_OEN_DCIO

Description: Each GMSL link has a built-in 8-bit idle packet error counter that counts detected idle packet errors. Idle packets have a data payload consisting of all zeros (before scrambling) and are transmitted to fill unused link bandwidth. The idle error counter increments when an 18-bit word in an idle packet is not equal to 0 after decoding and descrambling.

When the GMSL link is primarily idle, idle packets comprise most of the transmitted packets. If the link is run idle for an amount of time, checking the detected idle errors can provide information about the Bit Error Rate (BER) of the link during that period. Reference BER Testing using the GMSL2 Idle Link (PRBS Testing section) for additional information.

Note: An error is not detected by idle error detection if the error affects only the header of an idle packet.

Idle packet error counter registers:

- IDLE_ERR
- IDLE_ERR_A
- IDLE_ERR_B
- IDLE_ERR_C
- IDLE_ERR_D
- IDLE_ERR_SIO
- IDLE_ERR_DCIO

Each idle error flag is asserted when the corresponding idle error counter value is greater than or equal to the error counter threshold value defined by DEC_ERR_THR (default is 1).

Related registers: DEC_ERR_THR, AUTO_ERR_RST

Clearing: Idle error counters are reset to 0 when read, reset by the Auto Error Reset function, or after establishing link lock after loss of lock. After an idle error counter is reset, its error flag is also reset (the counter value of 0 is always less than the error threshold value).

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

33.3.3 GMSL Packet Count Interrupt

Note: the GMSL Packet Count Interrupt is intended for tests and diagnostics.

Flag:

- PKT_CNT_FLAG
- PKT_CNT_FLAG_A
- PKT_CNT_FLAG_B
- PKT_CNT_FLAG_C
- PKT_CNT_FLAG_D

OEN (Output Enable):

- PKT_CNT_OEN
- PKT_CNT_OEN_A
- PKT_CNT_OEN_B
- PKT_CNT_OEN_C
- PKT_CNT_OEN_D

Description: Each GMSL link has a built-in, user-configurable 8-bit received packet counter. The PKT_CNT_SEL and PKT_CNT_LBW registers are used to define the packet types that are counted. The counter is exponentially scaled. The user-configurable unit of count is defined as 2^N, where N is a value from 0 to 15 programmed with the PKT_CNT_EXP register.

Packet counter registers:

- PKT_CNT
- PKT_CNT_A
- PKT_CNT_B
- PKT_CNT_C
- PKT_CNT_D

The packet count flag is asserted when the packet counter value is greater than or equal to the packet counter threshold value defined by PKT_CNT_THR (default is 1).

Related registers: PKT_CNT_THR, PKT_CNT_EXP, PKT_CNT_SEL, PKT_CNT_LBW, AUTO_CNT_RST

Clearing: The packet counter is reset to 0 when read or reset by the Auto Packet Count Reset function. When a packet counter is reset, its error flag is also reset (the counter value of 0 is always less than the error threshold value).

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute Only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33.3.4 GMSL1 Error Flag

Flag:

- G1_A_ERR_FLAG
- G1_B_ERR_FLAG
- G1_C_ERR_FLAG
- G1_D_ERR_FLAG

OEN (Output Enable):

- G1_A_ERR_OEN
- G1_B_ERR_OEN
- G1_C_ERR_OEN
- G1_D_ERR_OEN

Description: The GMSL1 mode error flag is asserted due to one of these events:

- GMSL1 PRBS error
- GMSL1 Decoding error
- GMSL1 Eye Opening Monitor error
- GMSL1 Video Line CRC error
- GMSL1 Adaptive EQ under-boost error
- GMSL1 PKTCC mode maximum retransmission error
- GMSL1 PKTCC mode CRC error

Related registers: See GMSL1 receiver register block. See the GMSL1 Legacy Mode (Backward Compatibility) section for additional information.

Clearing: See related register descriptions.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

33.3.5 Line Fault Error

Flag:
- LFLT_INT

OEN (Output Enable):
- LFLT_INT_OEN

Description: Each GMSL device has a built-in Line Fault detector for link diagnostics. An error is reported when the line is shorted to power, shorted to ground, or the wires of the differential pair are shorted. An error is also reported when line is open. Refer to the Line Fault section for additional information.

Related registers: LF_0, LF_1, LF_2, LF_3, PU_LF_0, PU_LF_1, PU_LF_2, PU_LF_3

Clearing: Resolve the line error condition.

33.3.6 Eye Opening Monitor Error

Flag:
- EOM_ERR_FLAG_A
- EOM_ERR_FLAG_B
- EOM_ERR_FLAG_C
- EOM_ERR_FLAG_D
- EOM_ERR_FLAG_SIO
- EOM_ERR_FLAG_DCIO

OEN (Output Enable):
- EOM_ERR_OEN_A
- EOM_ERR_OEN_B
- EOM_ERR_OEN_C
- EOM_ERR_OEN_D
- EOM_ERR_OEN_SIO
- EOM_ERR_OEN_DCIO

Description: GMSL devices have a built-in Eye Opening Monitor that periodically measures the received serial data eye after equalization. An eye opening error is reported if the measured eye opening falls below the programmed threshold (EOM_MIN_THR). More information can be found in the “Eye Opening Monitor” section.

Related registers: EOM, EOM_DONE, EOM_EN, EOM_PER_MODE, EOM_CHK_THR, EOM_CHK_AMOUNT, EOM_MIN_THR

Clearing: The eye opening monitor is a latching bit. The error is set once a single eye opening measurement is below the programmed threshold and is cleared only when the flag is read. Note that if the last eye opening measurement is below the error threshold when the flag is read, the flag will not be cleared.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL Customers Under NDA

33.3.7 Lock Status Interrupt

Flag:
- LOCK
- LOCK_A
- LOCK_B

OEN (Output Enable):
- LOCK_OEN
- LOCK_A_OEN
- LOCK_B_OEN

Description: Flag is asserted when the GMSL link loses lock. This is useful if the LOCK pin cannot be monitored separately by the host controller.

Related registers: None

Clearing: When LOCK output to the ERRB pin is enabled, loss of lock causes ERRB to go low. ERRB goes high when the link locks if there are not any other active error drivers.

33.3.8 Loss of Lock Status Interrupt

Flag:
- LOSS_OF_LOCK_FLAG

OEN (Output Enable):
- LOSS_OF_LOCK_OEN

Description: Flag is asserted when the GMSL link loses lock. This flag is latching. Note that this interrupt is different than the live Lock Status Interrupt and is not cleared when the link locks again. This is useful if the LOCK pin cannot be monitored separately by the host controller.

Related registers: None

Clearing: Flag is cleared when read.

Note: Only some GMSL2 devices have this interrupt – check product specific datasheet for availability.

ADI Confidential for New Version under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

33.3.9 Retransmission Count Interrupt

Flag:

- RT_CNT_FLAG
- RT_CNT_FLAG_A
- RT_CNT_FLAG_B
- RT_CNT_FLAG_C
- RT_CNT_FLAG_D

OEN (Output Enable):

- RT_CNT_OEN
- RT_CNT_OEN_A
- RT_CNT_OEN_B
- RT_CNT_OEN_C
- RT_CNT_OEN_D

Description: Low-speed GMSL2 channels (i.e., control channels, PT_X, PT_Y, SPI, GPIO, Audio, Auto-HDCP) have a built-in Automatic Repeat Request / Automatic Retransmission (ARQ). When enabled, packets that are not acknowledged by the receiver are automatically retransmitted by the ARQ. Each low-speed channel has a 7-bit counter that tracks the number of ARQ retransmissions (i.e., RT_CNT register in the register block of each sub-channel). When a channel has at least one retransmission event and the retransmission count reporting is enabled (RT_CNT_OEN register in the register block of each sub-channel, disabled by default), then the combined RT_CNT_FLAG is asserted.

Related registers: The RT_CNT and RT_CNT_OEN registers of each sub-channel.

Clearing: The retransmission counter (RT_CNT) of each sub-channel is reset when it is read.

33.3.10 Maximum Retransmission Interrupt

Flag:

- MAX_RT_FLAG
- MAX_RT_FLAG_A
- MAX_RT_FLAG_B
- MAX_RT_FLAG_C
- MAX_RT_FLAG_D

OEN (Output Enable):

- MAX_RT_OEN
- MAX_RT_OEN_A
- MAX_RT_OEN_B
- MAX_RT_OEN_C
- MAX_RT_OEN_D

Description: Low-speed GMSL2 channels (i.e., control channels, PT_X, PT_Y, SPI, GPIO, Audio, Auto-HDCP) have a built-in Automatic Repeat Request / Automatic Retransmission (ARQ). When enabled, packets that are not acknowledged by the receiver are automatically retransmitted by the ARQ. If the same packet is retransmitted for a maximum number of times (set by MAX_RT register in the register block of each sub-channel) and the expected acknowledge packet is not received, then the maximum retransmission error of that sub-channel is set (MAX_RT_ERR register in the register block of each sub-channel). The MAX_RT_ERR flag of all sub-channels are combined to generate the combined maximum retransmission flag (MAX_RT_FLAG).

Related registers: MAX_RT and MAX_RT_ERR registers of each sub-channel.

Clearing: The maximum retransmission error (MAX_RT_ERR) of each sub-channel is reset when it is read.

Analog Confidential for NewVision under NDA
Maxim Integrated Company Confidential, Distribute to Original
Provided to GMSL Customers Under NDA

33.3.11 Remote Error

Flag:

- REM_ERR_FLAG
- REM_ERR_FLAG_A
- REM_ERR_FLAG_B

OEN (Output Enable):

- REM_ERR_OEN
- REM_ERR_OEN_A
- REM_ERR_OEN_B

Description: Each GMSL2 device can report the ERRB status of the remote-side serial link device if the GMSL link is locked. See detailed description in the Remote Error Reporting section.

Related registers:

ERR_TX_EN, ERR_TX_ID
ERR_TX_EN_A, ERR_TX_ID_A
ERR_TX_EN_B, ERR_TX_ID_B
REM_ERR_OEN, REM_ERR_FLAG
REM_ERR_OEN_A, REM_ERR_FLAG_A
REM_ERR_OEN_B, REM_ERR_FLAG_B
ERR_RX_EN, ERR_RX_ID, ERR_RX_RECVED
ERR_RX_EN_A, ERR_RX_ID_A, ERR_RX_RECVED_A
ERR_RX_EN_B, ERR_RX_ID_B, ERR_RX_RECVED_B
ERR_RX_EN_C, ERR_RX_ID_C, ERR_RX_RECVED_C
ERR_RX_EN_D, ERR_RX_ID_D, ERR_RX_RECVED_D
ERR_RX_EN2_A, ERR_RX_ID2_A, ERR_RX_RECVED2_A
ERR_RX_EN3_A, ERR_RX_ID3_A, ERR_RX_RECVED3_A
ERR_RX_EN4_A, ERR_RX_ID4_A, ERR_RX_RECVED4_A
ERR_RX_EN2_B, ERR_RX_ID2_B, ERR_RX_RECVED2_B
ERR_RX_EN3_B, ERR_RX_ID3_B, ERR_RX_RECVED3_B
ERR_RX_EN4_B, ERR_RX_ID4_B, ERR_RX_RECVED4_B

Reference device datasheet register descriptions for more detailed information.

Clearing: Resolve the error status on the remote device.

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

33.3.12 Video Line CRC Error

Flag:
- LCRC_ERR_FLAG

OEN (Output Enable):
- LCRC_ERR_OEN

Description: The video line CRC checker detects pixel corruption within video line packets with very high probability. Flag is asserted when a video line CRC error is detected. By default, the GMSL link has video line CRC checking enabled. Video line CRC is a 32-bit code appended to each video line and is transferred to the receiver via info frames. Reference the CRC Error Detection and ARQ Error Correction section for additional information.

Related registers: LCRC_ERR, LINE_CRC_EN, LINE_CRC_SEL

Clearing: The video line CRC flag is a latching bit that resets to 0 when read or the link establishes lock after loss of lock.

33.3.13 Video Pixel CRC Error

Flag:
- VID_PXL_CRC_ERR_FLAG

OEN (Output Enable):
- VID_PXL_CRC_ERR_OEN

Description: The video pixel CRC checker detects pixel corruption within video packets with very high probability. Flag is asserted when at least one video pixel CRC error is detected (VID_PXL_CRC_ERR > 0).

GMSL links have video pixel CRC checking disabled by default; video line CRC (enabled by default) provides sufficient coverage. Video pixel CRC is enabled by setting TX_CRC_EN = 1 in the video transmit register block for each video pipe in the GMSL2 serializer and RX_CRC_EN = 1 in the video receive register block for each video pipe in the GMSL2 deserializer. When enabled, each video packet carrying 36 pixels is protected by a 16-bit CRC value. Reference the CRC Error Detection and ARQ Error Correction section for additional information.

Related registers: TX_CRC_EN_VIDEO, RX_CRC_EN_VIDEO

Clearing: Video pixel CRC error counter is reset to 0 when it is read or due to Auto Error Reset function or due to link gaining lock after loss of lock. When video pixel CRC error counter is reset, its error flag is also reset.

ADI Confidential for New Version under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Confidential,
Provided to GMSL2_Customer UNDER NDA

33.3.14 Video Sequence Error

Flag:
- VID_SEQ_ERR

OEN (Output Enable):
- VID_SEQ_ERR_OEN

Description: Flag is asserted when the deserializer detects non-sequential video sequence numbers. This indicates errors in the video transmission, dropped video packets, and/or incorrect sequencing of packets combined from the two PHYs in dual-link mode. This flag is latching.

Related registers: None

Clearing: Flag is cleared when read.

33.3.15 Video Block Length Error

Flag:
- VID_BLK_LEN_ERR

OEN (Output Enable):
- VID_BLK_LEN_ERR_OEN

Description: Flag is asserted when the deserializer detects incorrect video packet block length. This flag is latching.

Related registers: None

Clearing: Flag is cleared when read.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

33.3.16 Video Mask Error

Flag:
- VIDEO_MASKED_0_FLAG
- VIDEO_MASKED_1_FLAG
- VIDEO_MASKED_2_FLAG
- VIDEO_MASKED_3_FLAG

OEN (Output Enable):
- VIDEO_MASKED_0_OEN
- VIDEO_MASKED_1_OEN
- VIDEO_MASKED_2_OEN
- VIDEO_MASKED_3_OEN

Description: Flag is asserted when video masking is active. Video masking means that the specific video pipe is outputting blank video lines (data = 0's, black pixels). Up to three video pipes can lose VIDEO_LOCK and data will continue to output.

Note: This error is only available on GMSL2 CSI-2 Quad Deserializers operating in synchronous aggregation modes (4WxH or Wx4H) and on video pipes 0-3.

Related registers: VIDEO_MASKED_OEN, VIDEO_MASKED_FLAG, MIPI_PHY_15, MIPI_PHY_16, MIPI_TX52

Clearing: The video_masked registers (0x934, 0x974, 0x9B4, 0x9F4) indicate whether a video pipe / MIPI output is currently masked or was previously masked. Video_Mask_Latch_Reset (0x8C7) resets the video_masked_latched registers (0x934, 0x974, 0x9B4, 0x9F4). Video pipe are individually configured to output status to the ERRB pin with VIDEO_MASKED_x_OEN ERRB output enable (0x49) and the associated flags (0x4A).

33.3.17 Video PRBS Error

Flag:
- VPRBS_ERR_FLAG

OEN (Output Enable):
- VPRBS_ERR_OEN

Description: Flag is asserted when the Video PRBS checker detects at least one error (VPRBS_ERR > 0) while the Video PRBS test is running. Reference the PRBS Testing section for additional information.

Related registers: VPRBS_CHK_EN, VPRBS_ERR

Clearing: The video PRBS error counter is reset to 0 and the flag is cleared when read.

Analog Devices Confidential – Distribution Only Under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Version under NDA

33.3.18 Audio PRBS Error

Flag:

- APRBS_ERR_FLAG
- APRBS_A_ERR_FLAG
- APRBS_B_ERR_FLAG
- APRBS_X_ERR_FLAG
- APRBS_Y_ERR_FLAG

OEN (Output Enable):

- APRBS_ERR_OEN
- APRBS_A_ERR_OEN
- APRBS_B_ERR_OEN
- APRBS_X_ERR_OEN
- APRBS_Y_ERR_OEN

Description: Flag is asserted when the Audio PRBS checker detects at least one error (APRBS_ERR > 0) while the Audio PRBS test is running. Reference the PRBS Testing section for additional information.

Related registers: APRBS_CHK_EN, APRBS_ERR

Clearing: The audio PRBS error counter is reset to 0 and the flag is cleared when read.

33.3.19 HDCP Interrupt

Flag:

- HDCP_INT

OEN (Output Enable):

- HDCP_INT_OEN

Description: When enabled, Auto-HDCP-1.3 can generate interrupts related to HDCP events. The AH_STATUS register indicates the event(s) generating the interrupt(s):

- Bit 0 = AH_FAIL – AHDCP operation error
- Bit 1 = AH_KSV_READY – Keys are updated and checked
- Bit 2 = AH_NEW_DEV – A new device has been connected
- Bit 3 = AH_ENCOFF – Encryption is not enabled for more than 0.5 second.
- Bit 4 = ENCRYPT_ON – Actively encrypting means KSV and key exchange has been successful
- Bits [7:5] = Auto HDCP State Machine status
  - 3'b000: AHDCP reset state
  - 3'b001: AHDCP wait video link to setup
  - 3'b010: AHDCP in first part of authentication
  - 3'b011: AHDCP in second/third part of authentication
  - 3'b100: V' not match
  - 3'b101: Max devices or cascade exceeded
  - 3'b110: R0 not match
  - 3'b111: Bad BKSV

Reference the HDCP section for additional information.

Related registers: AH_MASK_ENCOFF, H_MASK_NEWDEV, AH_MASK_KSVREADY, AH_MASK_FAIL, AH_STATUS

Clearing: Read AH_STATUS clear this interrupt.

Analog Devices Confidential, Distribution Only Under NDA
Maxim Integrated Confidential, Provided to GMSL Customers Under NDA
ADI confidential for review under NDA

33.3.20 HDCP2 Interrupt

Flag:
- HDCP2_INT_FLAG,
- HDCP22_INT_FLAG

OEN (Output Enable):
- HDCP2_INT_OEN,
- HDCP22_INT_OEN

Description: When Auto-HDCP-2.X is enabled, it can generate an interrupt due to certain HDCP-related events. The events related to registers sys_int_en and sys_int_stat are:
- bit 0: Authentication Failed
- bit 1: Topology Available
- bit 2: Topology Changed
- bit 3: Encryption Disabled

Reference the HDCP section for additional information.

Related registers: sys_int_en, sys_int_stat

Clearing: Read register sys_int_stat to clear the interrupt

33.3.21 Frame Sync Error

Flag:
- FSYNC_ERR_FLAG

OEN (Output Enable):
- FSYNC_ERR_OEN

Description: Flag is asserted when the Frame Sync error counter (FSYNC_ERR_CNT) exceeds the Frame Sync error threshold (FSYNC_ERR_THR). The Frame Sync block can continuously monitor the received VS signals and assert an error when all VS pulses are not within a certain window. Reference the Frame Sync section for additional details.

Related registers: FSYNC_ERR_CNT, FSYNC_ERR_THR, FSYNC_MODE, FRM_DIFF_ERR_THR, OVLP_WINDOW, EN_FSIN_LAST

Clearing: The Frame Sync error counter is reset to 0 when it is read. When the error counter is reset, its error flag is also reset (the counter value of 0 is always less than the error threshold value).

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_ Customers Under NDA

33.3.22 Watermark Detect Interrupt

Flag:
- WM_ERR_FLAG

OEN (Output Enable):
- WM_ERR_OEN

Description: Flag is asserted when a watermark detection error is detected. The flag is latching. More information can be found in the Watermarking section.

Related registers: WM_EN, WM_DET, WM_ERROR, WM_DETOUT

Clearing: The watermark error flag is cleared when it is read. It will not be cleared if watermark is still not detected (i.e., watermark error condition) when the flag is read. WM_ERROR shows the live (non-latching) detection status.

33.3.23 DSC Encoder Self-Test Error

Flag:
- dsce_selftest_err

OEN (Output Enable):
- dsce_err_oen

Description: Refer to the DSC Encoder Self-Test section for more details of this error.

Related registers: dsce_selftest_err_flag, dsce_selftest_done, dsce_selftest_err, dsce_selftest_mb_done, dsce_selftest_mb_fail

Clearing: This status flag is cleared when DSC_STATUS at 0x25E is read.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

33.3.24 DSC Decoder Error

Flag:
- DSCD_ERR_FLAG

OEN (Output Enable):
- DSCD_ERR_OEN

Description: This flag indicates an error condition within the DSC decoder. It is a surrogate for multiple DSC error events that are individually enabled and ORed together. The flag is latched on the first assertion of an enabled event and will not be retriggered until the underlying events interrupt status bits are cleared (see Table 152).

Related registers: The DSC decoder block contains the interrupt status register bits and corresponding interrupt enable register bits that record certain error events in the operation of the DSC decoder. The interrupt status bits are latching bits that remain set until the bits are written to 1. Events are always recorded by the status bits even if the interrupt enable bits are not set. When the interrupt enable bit are set, the corresponding interrupt status events are ORed together and used to trigger the DSDC_ERR_FLAG.

Table 152. DSC Decoder Error Related Registers

Interrupt Status/Enable	Description
vb_ovfl_ist/vb_ovfl_ien	Video buffer overflow interrupt status/enable
pb_ovfl_ist/pb_ovfl_ien	Pixel buffer overflow interrupt status/enable
ib0_ovfl_ist/ib0_ovfl_ien	Input buffer 0 overflow interrupt status/enable
ib0_undfl_ist/ib0_undfl_ien	Input buffer 0 underflow interrupt status/enable
tc_size0_err_ist/tc_size0_err_ien	TC size 0 error interrupt status/enable
rcb0_ovfl_ist/rcb0_ovfl_ien	RC buffer 0 overflow interrupt status/enable
ib1_ovfl_ist/ib1_ovfl_ien	Input buffer 1 overflow interrupt status/enable
ib1_undfl_ist/ Ib1_undfl_ien	Input buffer 1 underflow interrupt status/enable
tc_size1_err_ist/tc_size1_err_ien	TC size 1 error interrupt status/enable
rcb1_ovfl_ist/rcb1_ovfl_ien	RC buffer 1 overflow interrupt status/enable

Clearing: This flag is cleared when read. The flag will not retrigger even if the underlying condition is persistent unless the underlying events interrupt status bits within the DSC decoder block are cleared (see Related Registers and Table 152).

ADI Confidential for review/sign under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customer_Name_NDA

33.3.25 FEC Corrected and Uncorrectable Errors

Flag:
- FEX_RX_ERR_FLAG

OEN (Output Enable):
- FEC_RX_ERR_OEN

Description: Flag is asserted when the GMSL Forward Error Correction (FEC) decoder's uncorrected error counter (UNCORRECTED_BLOCKS) exceeds uncorrected error threshold (UNCORRECTED_ERROR_THRESHOLD) or the corrected error counter (BIT_ERRS_CORRECTED) exceeds corrected error threshold (BIT_ERRS_CORRECTED_THRESHOLD).

Related registers: UNCORRECTED_BLOCKS, UNCORRECTED_ERROR_THRESHOLD, BIT_ERRS_CORRECTED, BIT_ERRS_CORRECTED_THRESHOLD

Clearing: Write 1 to CLEAR_STATS, CLEAR_BLOCKS_UNCORRECTABLE or CLEAR_BITS_CORRECTED.

33.3.26 RGBA Split Error

Flag:
- rgba_split_err_flag

OEN (Output Enable):
- rgba_split_err_oen at 0x25F bit 4

Description: The RGBA_SPLIT block, after the HDMI Rx block in the serializer, detects error condition(s) in the RGBA splitter operation (e.g., video horizontal line length exceeds the maximum 5760 pixels). The flag is latching.

Related registers: None

Clearing: This error flag is cleared by rewriting a 1 to rgba_split_en at 0x1400 and reading register DSC_STATUS (register 0x025E).

33.3.27 RGBA Combiner Error

Flag:
- rgba_comb_err_flag

OEN (Output Enable):
- rgba_comb_err_oen

Description: The RGBA combiner block in the GMSL2 serializer combines the Alpha character previously stripped off by the RGBA_SPLIT block with the DSC-encoded byte. This block can detect a combining operation error condition. This flag is latching.

Related registers: None

Clearing: This error flag is cleared by rewriting a 1 to rgba_comb_en at 0x1400 and reading register DSC_STATUS (register 0x025E).

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33.3.28 ADC Interrupt

Flag:
- ADC_INT_FLAG

OEN (Output Enable):
- ADC_INT_OEN

Description: ADC interrupt. Asserted whenever any monitored ADC interrupt is asserted. Refer to the ADC section for additional information.

Related registers: ADC_INTR0→ADC_INTR3, ADC_LIMITX_0→ADC_LIMITX_3 (where X can be 0→7).

Clearing: The flag is cleared when the respective latched error source register is read.

33.3.29 VDD Comparator Error

Flag:
- VDDCMP_INT_FLAG

OEN (Output Enable):
- VDDCMP_INT_OEN

Description: Flag is asserted when the internal voltage comparator detects that selected internal VDD node is below a threshold. See CMP_STATUS description. This flag is latching. Refer to the Power Manager and Sleep Mode section for additional information.

Related registers: CMP_STATUS, CMP_STATUS_MASK, VDDCMP_MASK

Clearing: Flag is cleared when read.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33.3.30 VDD Brown Out Error

Flag:
- VDDBAD_INT_FLAG

OEN (Output Enable):
- VDDBAD_INT_OEN

Description: Flag is asserted when the internal voltage comparator detects either VDD or VDDA has dropped below 0.82V. This flag is latching. Refer to the Power Manager and Sleep Mode section for additional information.

Related registers: VDDBAD_STATUS

Clearing: Flag is cleared when read.

33.3.31 PORZ VDD Status Interrupt

Flag:
- PORZ_INT_FLAG

OEN (Output Enable):
- PORZ_INT_OEN

Description: Flag is asserted when the internal voltage comparator detects that VDD18 is below 1.516V during power-up. This flag is latching. Refer to the Power Manager and Sleep Mode section for additional information.

Related registers: PORZ_STATUS

Clearing: Flag is cleared when read.

33.3.32 VDD Over-Voltage Interrupt

Flag:
- VDD_OV_FLAG

OEN (Output Enable):
- VDD_OV_OEN

Description: Flag is asserted when the internal voltage comparator detects that internal VDD, VDD1P8, or VREG voltage is above the threshold set by OV_LEVEL, OV1P8_LEVEL, or OVREG_LEVEL registers and the video channel is active (locked). Refer to the Power Manager and Sleep Mode section for additional information.

Related registers:
VDD_OV_LIVE, OV_LEVEL, OV_HYSTERESIS, VDD1P8_OV_LIVE, OV1P8_LEVEL, OV1P8_HYSTERESIS, VREG_OV_LIVE, OVREG_LEVEL, OVREG_HYSTERESIS

Clearing: Flag is cleared when read.

ADI Confidential for New Vision under NDA
Analog Devices Confidential – Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

33.3.33 HDMI Interrupt (HDMI Serializers)

Flag:
- HDMI_INT

OEN (Output Enable):
- HDMI_INT_OEN

Description: GMSL2 HDMI serializers can detect certain HDMI events and generate interrupts. This is driven by the intern HDMI Rx IP. For more information, refer to HDMI Interrupts in the HDMI Serializers section.

Related registers: Refer to the HDMI Interrupts in the HDMI Serializers section.

Clearing: Refer to the HDMI Interrupts in the HDMI Serializers section.

33.3.34 HDMI CRC Error

Flag:
hdmi_crc_err_flag

OEN (Output Enable):
hdmi_crc_err_oen

Description: Video frame CRC can be generated at the output of the HDMI Rx block and checked prior to the DSC encode operation by writing 1 to hdmi_crc_err_oen. The error condition caused by nonmatching CRC values is routed to the ERRB pin.

Related registers: DSC_STATUS, DSC_CTRL

Clearing: This error flag is cleared when DSC_STATUS at 0x25E is read.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

33.3.35 OLDI Output Stuck Error

Flag:
- OUT_STUCK_ERR_FLAG

OEN (Output Enable):
- OUT_STUCK_ERR_OEN

Description: Flag is asserted when the OLDI output is stuck either low or high.

Related registers: None

Clearing: The flag is cleared when read if the output stuck event has been resolved.

33.3.36 OLDI Output Open Error

Flag:
- OUT_OPEN_ERR_FLAG

OEN (Output Enable):
- OUT_OPEN_ERR_OEN

Description: Flag is asserted when the OLDI output is open.

Related registers: None

Clearing: The flag is cleared when read if the output open event has been resolved.

33.3.37 MIPI Rx Error

Flag:
- MIPI_ERR_FLAG

OEN (Output Enable):
- MIPI_ERR_OEN

Description: Flag is asserted when MIPI Rx controller detects an error.

Related registers: MIPI_RX12→MIPI_RX20

Clearing: The flag is cleared when the respective latched error source register is read.

ADI Confidential for New vision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33.3.38 PMX Error

Flag:
- PMX_ERR_FLAG

OEN (Output Enable):
- PMX_ERR_OEN

Description: This flag indicates that a status event (including error conditions) was updated by the DisplayPort (DP) link management PicoMax (PMX) microcontroller. It is a surrogate for PMX_SS_STATE_B1 register, bit 7. The flag is latched on the first assertion of PMX_SS_STATE_B1 bit 7 and will not be retriggered until the PMX_SS_STATE_B1 bit 7 is cleared (see Related Registers).

Related registers: The PicoMax microcontroller maintains status for DP link management (i.e., training, video control, etc.). The encoded status is stored in the 2 bytes of PMX_SS-PMX_SS_STATE_B0/1. When the status is updated by the PicoMax, it sets bit 7 of PMX_SS_STATE_B1 to indicate that the new status is valid, which then sets PMX_ERR_FLAG. For proper operation, only the PicoMax should read/write PMX_SS-PMX_SS_STATE_B0/1. The normal control channel can read MISC-PMX_SS_STATE_B0/1 to retrieve the status and clear the valid bit (PMX_SS_STATE_B1, bit7). The meaning of the encoded status is defined in other documents.

Table 153. PMX Error Related Registers

Register	Bit	Description
MISC	PMX_SS_STATE_B0	Read only version of PMX_SS-PMX_SS_STATE_B0
MISC	PMX_SS_STATE_B1	Read only version of PMX_SS-PMX_SS_STATE_B1. Reading this register clears bit 7 which is used to set PMX_ERR_FLAG.
PMX_SS	PMX_SS_STATE_B0	Low byte of PicoMax subsystem state for DP link management.
PMX_SS	PMX_SS_STATE_B1	High byte of PicoMax subsystem state for DP link management. Bit 7 is used to set PMX_ERR_FLAG.

Clearing: This flag is cleared when read. The flag will not retrigger even if the underlying condition is persistent unless the underlying event is cleared (see Related Registers).

ADI Confidential for Review under NDA
Analog Devices Confidential Property of Maxim Integrated Company Confidential Provided to GMSL2_Customers under NDA

33.3.39 Internal Memory Error

Flag:
- MEM_INT_ERR_FLAG

OEN (Output Enable):
- MEM_INT_ERR_OEN

Description: Some GMSL2 devices can continuously monitor internal memories for errors and assert this flag when an error is detected. The flag is latching. When a memory error is detected, the host SoC can decide to discard the transmitted video frame.

Related registers: DV_MEM_CRC_ERR, DV_MEM_CRC_ERR_A, DV_MEM_CRC_ERR_B, BACKTOP_MEM_CRC_ERR

Clearing: The flag is cleared when MEM_INT_ERR_FLAG is read.

33.3.40 Retention Memory CRC Error

Flag:
- RTTN_CRC_INT

OEN (Output Enable):
- RTTN_CRC_ERR_OEN

Description: Flag is asserted when a CRC error is detected after coming out of sleep mode (after reading retention memory).

Related registers: INJECT_RTTN_CRC_ERR

Clearing: This flag is cleared when read.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

33.3.41 eFuse CRC Error

Flag:

- EFUSE_CRC_INT

OEN (Output Enable):

- EFUSE_CRC_ERR_OEN

Description: When the EFUSE is programmed during the device's production testing, a CRC signature is added to the last two locations of the EFUSE array. This signature is generated using the programming data for the remainder of the EFUSE array. On power-up, the contents of the EFUSE are read by the device, and its contents are used to initialize and configure device specific registers. As part of this initial read, the device recomputes the CRC based on the read values and checks it against the signature programmed into the array. If this signature is incorrect, the contents of the EFUSE have changed since the initial programming and the device asserts the ERRB flag to signal an issue.

Related registers: INTR6, INTR7

Clearing: This flag is cleared when read.

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

33.4 Debug Techniques

The error generator function (ERRG_EN) can be used to evaluate a device's error detection and reporting capabilities. The error generator injects bit or burst errors to the outgoing link (i.e., the forward channel in serializer and the reverse channel in deserializers). This function can be used to evaluate the decoding error (DEC_ERR), idle error (IDLE_ERR), retransmission count (RT_CNT), maximum retransmission error (MAX_RT), Audio PRBS, and Video PRBS error detection and reporting.

33.5 Feature Availability by Device Family

ERRB is supported by all GMSL2 devices. Not all errors/interrupts addressed in this section are available in all devices. Reference device-specific datasheets for available errors and interrupts. If the register controlling error/interrupt output to the ERRB pin is not listed, that error/interrupt reporting function is not available in that device.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

34 CRC Error Detection and ARQ Error Correction

34.1 Overview

GMSL2 serial links incorporate 16-bit Cyclic Redundancy Check (CRC) protection for error detection of control channel (including I2C, UART, SPI, and GPIO), video, and audio data. The 16-bit CRC can also include Automatic Repeat Request (ARQ) error correction on both the forward and reverse channels (not available for RGMII or video data). Video data is protected with 32-bit CRC error detection by default.

Note: Forward Error Correction (FEC) is used in GMSL2 devices to detect and correct bit errors occurring during the transmission of compressed video on the serial link. Reference the Forward Error Correction (FEC) for additional details.

CRC ensures link errors caused by EMI or other noise events do not corrupt control channel data.

Note: Video data can be protected with either 16-bit or 32-bit CRC error detection. See Video Data CRC for more information.

34.2 CRC Operation

34.2.1 16-bit CRC

Every video, audio, and control channel packet (excluding idle and acknowledge packets) can be protected with 16-bit CRC. For enabled channels, the CRC block generates a 16-bit code (calculated with the polynomial below) that is appended to each packet. All low-bandwidth control channels have packet CRC enabled by default; each packet type can be individually configured to enable or disable CRC protection. The main and pass-through control channels are protected in both I2C and UART modes.

The 16-bit packet CRC generator calculates following polynomial:

$$CRC16 = x^{16} + x^{15} + x^{2} + 1$$

Note: This is the same polynomial used in the USB protocol for data packets.

34.2.2 32-bit CRC

Video data can be protected with a 32-bit CRC calculated per video line. The CRC block generates a 32-bit code that is appended to each DE (default) or HS (selected by register) pulse. This code is transferred to the receiver side of the serial link via info frames. The receiver-side CRC checker generates the same code and checks if the CRC codes match. An error is asserted if the codes do not match. The CRC check is processed at every falling edge of DE (or HS) on the receiver side even if the info frame is not received.

The 32-bit packet CRC generator calculates the following polynomial:

$$CRC32 = x^{32} + x^{31} + x^{28} + x^{25} + x^{24} + x^{23} + x^{21} + x^{18} + x^{11} + x^{8} + x^{7} + x^{6} + x^{5} + x^{3} + x^{1} + 1$$

ADI Confidential for Review under NDA
Maxim Integrated Confidential, Provided for GMSL2 Customers under NDA

34.2.3 Video Data CRC

Video data can be protected with either 16-bit or 32-bit CRC. These are two different error detection schemes: Video Pixel CRC (16-bit) and Video Line CRC (32-bit). Video Pixel CRC detects pixel corruption in each packet of video data (36 pixels) with a 16-bit CRC value. Video Line CRC detects pixel corruption in each line of video data with a 32-bit CRC value. By default, Video Line CRC is enabled, and Video Pixel CRC is disabled.

Video Line CRC (default) provides robust video data protection with minimal bandwidth overhead. However, there are use cases where Video Pixel CRC may be the preferred video data CRC scheme:

- Applications that require errors to be detected as soon as possible.
- Applications where the rough location of the error in the line would like to be known.

34.3 Automatic Repeat Request / Automatic Retransmission (ARQ)

Automatic Repeat Request (ARQ) is an automatic packet retransmission method used to ensure data integrity on communications channels with low-bandwidth control data. ARQ works in conjunction with 16-bit packet CRC to detect whether packets are received without error or not. With a successful data transmission, the ARQ mechanism on the transmit side receives confirmation of error-free transmission from the receiver side. In the case of a transmission error (e.g., corrupted or dropped packet), the ARQ on the transmit side does not receive confirmation of an error-free transmission and automatically retransmits the packet.

34.3.1 ARQ Operation

Each control packet is appended by a 4-bit sequence number that continuously increments and rolls over. The transmitter saves the last 15 packets transmitted on each communication channel. When the receiver receives a control packet, it checks that the CRC and sequence number are correct. If both are validated, the receiver sends an acknowledgement packet back to notify that the packet with a certain sequence number has been correctly received. If the transmitter does not get an acknowledge packet from the receiver for a data packet with a certain sequence number, it automatically retransmits the data packet after reading it from the internal packet memory.

The 4-bit sequence number allows the transmitter to transmit up to 15 packets without receiving acknowledgment from the receiver. This allows pipelined operation which minimizes the effects of round-trip latency with acknowledge packets and increases the continuous available bandwidth.

The acknowledge packet uses the same header field as low-bandwidth packets, but it begins with a different special symbol to distinguish it from regular data packets. This simplified format keeps retransmission exchanges independent from the communication channel. Note that this smaller packet format contains no data, obviating the need for full 16-bit CRC. Instead, acknowledge packets are protected by a 5-bit CRC (polynomial: $x^5 + x^2 + 1$). The acknowledge packets include the same 4-bit sequence number of the correctly received data packet.

ADI Confidential for Review Only
Analog Devices Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

34.3.1.1 Go-Back-N

The GMSL2 ARQ mechanism operates using the 'Go-Back-N' principle, where N = 15 for a 4-bit sequence number. This 15-packet sliding window improves bandwidth usage and availability as 15 packets can be transmitted without receiving an acknowledge.

Figure 130 below demonstrates 'Go-Back-N' where N = 3.

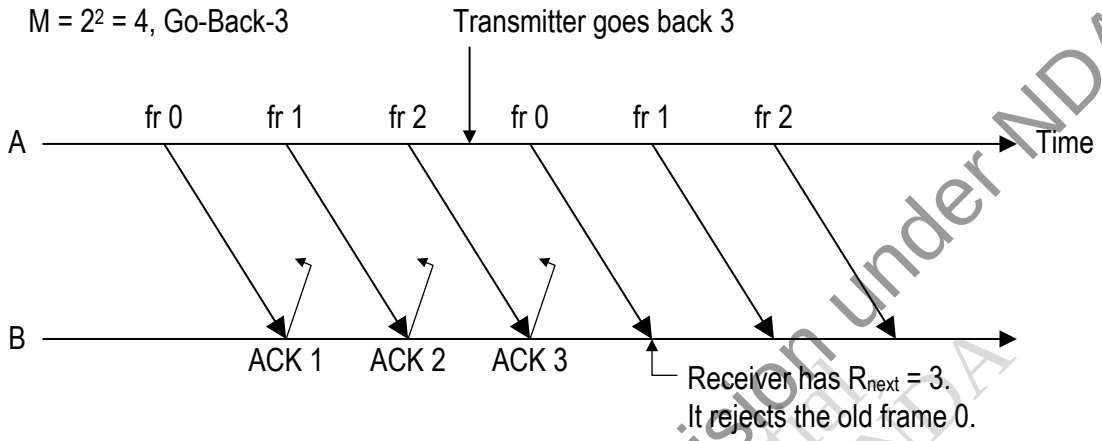


Figure 130. Operation of Go-Back-N ARQ with Go-Back-3

In this example, the transmitter sends up to three packets. The transmitter begins waiting for the acknowledge after the first packet is sent; if more than one packet is sent, waiting and transmission of the additional packets is handled in parallel. If the expected acknowledge does not arrive, the transmitter goes back three and retransmits the packet. Any retransmission attempt is counted and stored. This retransmission process repeats until the package is either acknowledged or the maximum number of retransmissions is reached. Upon reaching the maximum, the packet is assumed to be acknowledged, it is dropped from the transmission queue, and the send window moves forward to transmit subsequent packets. In cases where the maximum number of retransmissions is met, a register error is flagged and the ERRB pin changes state to signal 'Error Output.'

Note: In the actual GMSL2 ARQ system, N = 15.

A 3-bit register is allocated to set the maximum number of retransmissions. A packet can be retransmitted up to seven times depending on the register value set in MAX_RT for each communication channel.

The receiver side has a receiver window of size one and only waits for one packet with a match to the expected sequence number. The receiver passes a packet to the corresponding adapter if the packet is received completely without error and the sequence number matches the expected value. Here, the receiver also sends an acknowledge packet back to the transmitter. In the case of an error or unmatched sequence number, the receiver will drop the received packet.

Analog Confidential for New Division under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

34.3.1.2 ARQ Path

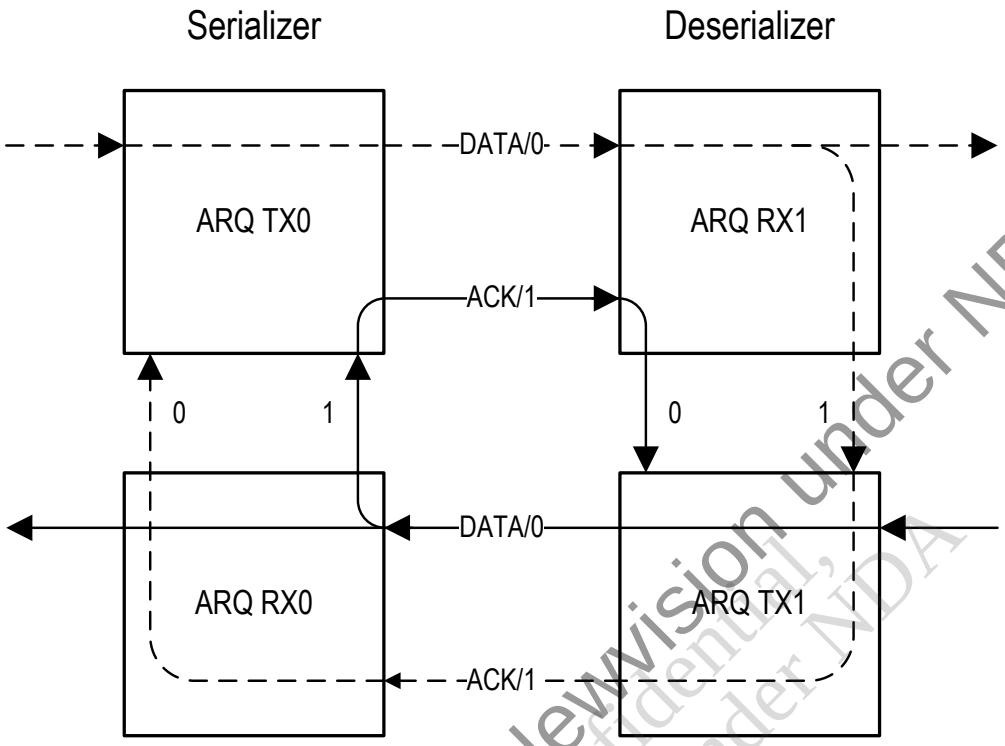


Figure 131. ARQ Path

The dashed line in the Figure 131 shows the data flow from the serializer to the deserializer and the subsequent acknowledge from the deserializer to the serializer. The solid line depicts the data flow from the deserializer to the serializer and the subsequent acknowledge from the serializer to the deserializer. The ARQ blocks of both the serializer and the deserializer can operate simultaneously and support bidirectional data transfer(s). For example, ARQ TX0 (in the serializer) can transmit either data or an acknowledge of data received from the deserializer. The same function is performed by ARQ TX1 in the deserializer. Here, ARQ TX1 can transmit either data or an acknowledge of data received from the serializer. The ARQ RX blocks operate in a similar manner with the corresponding ARQ TX blocks. This coordination facilitates bidirectional operation.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Confidential, Provided to GMSL Customers Under NDA

34.4 CRC/ARQ Configuration

The following sections describe how to enable/disable and configure CRC and ARQ in GMSL2 devices. CRC is enabled by default for all packet types except RGMII; ARQ is enabled by default for all supported packet types. ARQ does not support RGMII or video.

34.4.1 32-bit CRC Configuration

To enable Video Line CRC, set the bit LINE_CRC_EN to 1 in the serializer for each video pipe that needs CRC protection. Additionally, set LINE_CRC_EN to 1 for the deserializer video pipe that is receiving the video. This is enabled in all serializers and deserializers by default.

The CRC code can be appended to either the DE or HS pulse. This is configured with the LINE_CRC_SEL bit. Write 0 to use the DE pulse for video line CRC and write 1 to use the HS pulse.

34.4.1.1 32-bit Video CRC Errors

Video Line CRC errors are reported to the LCRC_ERR_FLAG bit in the deserializer register INTR7. To enable/disable the output of this error flag to the ERRB pin, set the LCRC_ERR_OEN bit in the deserializer register INTR6.

34.4.2 16-bit CRC Configuration

To enable packet CRC for the audio or control channels, set the bit TX_CRC_EN to 1 in the serializer for each packet type that needs CRC protection. Additionally, set RX_CRC_EN to 1 for the corresponding deserializer packet types. This is enabled for all audio and control channel packet types, except RGMII, by default.

Table 154 lists the packet types that can enable/disable 16-bit CRC with their corresponding registers.

Table 154. 16-bit CRC Registers

16-bit CRC Packet Type	Register Block*	Register Name
RGMII†	RGMII	TR0
Video X†	VIDEO_X	TX0
Video Y†	VIDEO_Y	TX0
Audio X	AUDIO_X	TR0
Audio Y	AUDIO_Y	TR0
SPI	SPI	TR0
GPIOs	GPIO	TR0
HDCP Control Packets	AHDCP	TR0
Main Control Channel‡	CC	TR0
Pass-through Channel 1	IIC_X	TR0
Pass-through Channel 2	IIC_Y	TR0

* Register block name may vary by device family. Reference device datasheet for specific register block name. For example, devices with only one audio block, the audio registers are named AUDIO without "_X" or "_Y" suffix.
† Not enabled by default. Use of CRC on RGMII packets is discouraged as RGMII has error detection and retransmission mechanisms at a higher protocol level. Video Line CRC should be disabled if Video Pixel CRC is enabled.
‡ CRC and ARQ should not be disabled on the main control channel packets.

Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA
API Confidential for New Vision under NDA

Note: CRC configuration for pass-through channel 1 (in I2C or UART mode) is controlled by register IIC_X_TR0 and CRC configuration for pass-through channel 2 (in I2C or UART mode) is controlled by register IIC_Y_TR0.

34.4.2.1 16-bit Video CRC Errors

Video Pixel CRC errors are reported to the VID_PXL_CRC_ERR_FLAG bit in the deserializer register INTR7. To enable/disable the output of this error flag to the ERRB pin, set the VID_PXL_CRC_ERR_OEN bit in the deserializer register INTR6.

34.4.3 16-bit ARQ Configuration

The ARQ configuration registers for each channel are listed below in Table 155. By default, ARQ is enabled for all channels (except RGMII and Video Pixel), and error reporting to the ERRB pin is enabled.

Note: ARQ is not available for 16-bit Video Pixel CRC or RGMII.

Table 155. 16-bit ARQ Registers

Control Channel Packet Type	Register Block*	Register Names
RGMII†	Not available	Not available
Video X†	Not available	Not available
Video Y†	Not available	Not available
Audio X	AUDIO_X	ARQ0 – ARQ2
Audio Y	AUDIO_Y	ARQ0 – ARQ2
SPI	SPI	ARQ0 – ARQ2
GPIOs	GPIO	ARQ0 – ARQ2
HDCP Control Packets	AHDCP	ARQ0 – ARQ2
Main Control Channel‡	CC	ARQ0 – ARQ2
Pass-through Channel 1	IIC_X	ARQ0 – ARQ2
Pass-through Channel 2	IIC_Y	ARQ0 – ARQ2

* Register block name may vary by device family. Reference device datasheet for specific register block name.
† ARQ is not available.
‡ CRC and ARQ should not be disabled on the main control channel packets.

Note: ARQ configuration for pass-through channel 1 (in I2C or UART mode) is controlled by register IIC_X_ARQ0 – ARQ2 and ARQ configuration for pass-through channel 2 (in I2C or UART mode) is controlled by register IIC_Y_ARQ0 – ARQ2.

If there is a CRC error, the corrupted packet will be retransmitted as described in the *ARQ Operation* section above.

AD Confidential for New Vision under NDA
Analog Devices Confidential for Maxim Integrated Company Provided to GMSL2 Customers Under NDA

34.4.3.1 Reporting of CRC Errors/ARQ retries

ARQ can only be enabled on channels with CRC enabled. If any of the CRC-enabled control channels reports an error, the corrupted/dropped packet is retransmitted with the ARQ scheme. If the bit RT_CNT_OEN (register xx_ARQ1) is set high for a given channel, the flag bit RT_CNT_FLAG (register INTR5) will be set high; if the main RT_CNT_OEN (in register INTR4) is enabled, the error will be reported to the ERRB pin.

The number of ARQ retries is reported to RT_CNT[6:0] (register xx_ARQ2) for each channel.

34.4.3.2 Reporting of Maximum ARQ retries

If the number of retries for a channel exceeds a threshold (set by MAX_RT[2:0]), it is reported to MAX_RT_FLAG. If the MAX_RT_ERR_OEN (register xx_ARQ1) is set high, the error will be reported to the main MAX_RT_FLAG (register INTR5) error bit. If MAX_RT_OEN (register INTR4) is set high, the error will be reported to the ERRB pin.

34.5 Availability by Device Family

32-bit CRC, 16-bit CRC, and ARQ are available on all GMSL2 devices (Table 156).

Table 156. CRC/ARQ Availability by Device Family

Device Family	32-bit CRC	16-bit CRC	ARQ
HDMI Serializers	X	X	X
Advanced HDMI Serializers	X	X	X
DSC Serializers	X	X	X
Advanced DSI Serializers	X	X	X
CSI-2 Serializers	X	X	X
Advanced CSI-2 Serializers			
CSI-2 Camera Deserializers	X	X	X
CSI-2 Display Deserializers	X	X	X
OLDI Deserializers	X	X	X
eDP/DP Deserializers	X	X	X
CSI-2 Quad Deserializers	X	X	X

ADI Confidential for New Version, under NDA
Analog Devices Confidential, Distribution only with NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

35 Voltage Monitoring

35.1 Overview

GMSL2 devices monitor various onboard supply voltages and provide alerts for over- or undervoltage conditions. Dedicated status register flags are driven by internal comparators that measure each supply voltage relative to an internal voltage reference. These registers are latching status flags. Supply voltages with latching status flags retain alerts and can be used to capture temporary conditions and be read back later (at which time the previously latched state is cleared). Many status flags have user-configurable parameters including custom voltage thresholds and the option to report status registers to the ERRB pin (which will assert low when an errant condition is sensed). Reference the GMSL2 Error Reporting (ERRB Pin) section for additional information.

Note: Each GMSL2 device has a unique combination of voltage monitoring and status/error reporting mechanisms. Reference *Feature Availability by Device Family* for details. A small subset of the devices provides additional power supply monitoring capabilities via an integrated ADC. Refer to the *ADC* section for further information regarding this function.

35.1.1 Architecture

Most GMSL2 devices include a common set of power supplies that provide power to the digital core (VDD_sw), GMSL link circuitry (VDD18), and general GPIO pins (VDDIO). All devices include undervoltage monitoring on the common set of power supplies with some providing extended monitoring capabilities. Each GMSL2 product family has unique, family-specific power supplies with dedicated functionality. Examples here include HDMI power (VDD33), RGMII power (VDDIORG), and MIPI output power (VTERM). Except for VTERM, these family-specific power supplies are not typically monitored.

The following table details the power supplies for which voltage monitoring is available (Table 157).

Table 157. Power Supply Monitoring Functions Available

Supply Name	Description	Monitor Functions Available
VDD_sw	Internal 1V digital core supply	Undervoltage (all devices) Overvoltage (some devices)
VDD18	GMSL 1.8V supply	Undervoltage (all devices)
VDDIO	1.8V to 3.3V I/O supply	Undervoltage (all devices)
VTERM	1.2V MIPI CSI-2 Output Supply	Undervoltage (CSI-2 deserializers only)
VDD/VREG	1V LDO input	Not monitored
VDD33, VDDA, VDDIORG	Misc. special function supplies	Not monitored

ADI Confidential for review under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

The following block diagram (Figure 132) details the connectivity between the voltage monitor and internal power supplies. The naming convention detailed here is representative of most GMSL2 devices. In some cases, there may be dedicated analog VDD pins and/or explicit VDD regulator input pins (e.g., VREG). Note that VDD/VREG pins are not typically monitored.

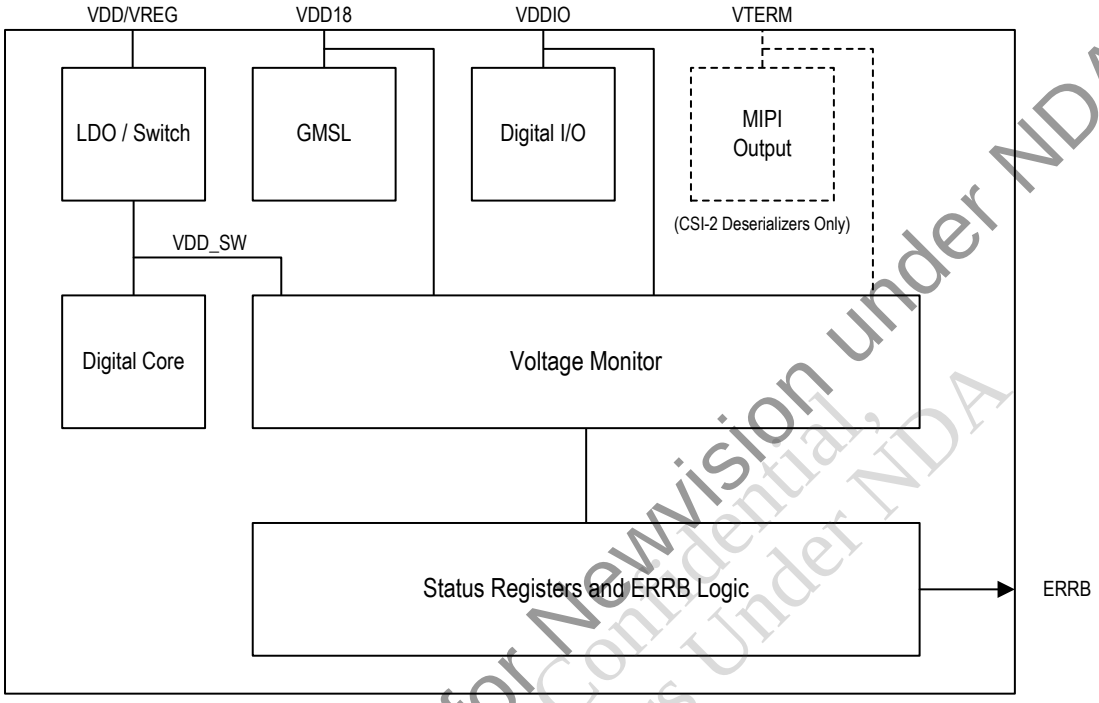


Figure 132. Voltage Monitor and Internal Power Pins

ADI Confidential for Newvision under NDA
Analog Devices Confidential,
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

35.2 Operation

35.2.1 VDD_sw Monitoring Details

Undervoltage (UV) and overvoltage (OV) monitoring of VDD_SW (the primary 1V core supply) is included on nearly all GMSL2 devices (VDD_sw OV monitoring is not available on HDMI and advanced HDMI serializers). VDD_sw is typically derived from VDD/VREG either via an internal LDO or series switches.

35.2.1.1 Undervoltage Monitoring of VDD_SW

An undervoltage condition on VDD_sw nominally occurs when VDD_sw < 0.82V. Note that the precise threshold varies some between devices. Refer to the datasheet specific to the part number to verify the specified VDD_sw undervoltage threshold for a given device.

In the case of an undervoltage condition, the power manager triggers a reset of the digital core and resets all registers powered by VDD_sw. When power has recovered, the reset is released. The reset of the digital core ensures that a brownout does not result in corruption of the registers.

The presence of a undervoltage event is recorded via two status flags:

- VDDBAD_STATUS[1] and VDDBAD_STATUS[0] – latched high following undervoltage event.
- CMP_STATUS[2] – latched low following undervoltage.

The memory that maintains the VDDBAD_STATUS and CMP_STATUS[2] bits is powered by the 1.8V power supply; as a result, they are persistent following a reset triggered by a VDD_sw undervoltage event. This enables a VDD_sw undervoltage event to be reported following the recovery of the VDD power supply, and the associated interrupt flags, VDDBAD_INT_FLAG and VDDCMP_INT_FLAG, can drive ERRB low to alert the system of a brownout.

To clear VDDBAD_INT_FLAG, the associated VDDBAD_STATUS bits must be read first to clear. After the VDDBAD_STATUS bits are cleared, VDDBAD_INT_FLAG can then be read, at which point the flag will be cleared. The process to clear VDDCMP_INT_FLAG is similar (i.e., the associated CMP_STATUS bit must be cleared first via reading).

35.2.1.2 Overvoltage Monitoring of VDD_SW

An overvoltage condition on VDD_sw is reported if the observed voltage exceeds a user-selectable threshold specified by the associated OV_LEVEL bit field. Refer to the datasheet specific to the part number for details regarding the specific threshold levels supported by a given device. If VDD_sw is greater than the specified threshold, the VDD_OV_FLAG is set. The VDD_OV_FLAG is a latching bit that is set when an overvoltage condition occurs and does not clear until read. VDD_OV_FLAG can be configured to drive the ERRB pin to alert the system of a fault condition. Note that operation of the VDD_OV_FLAG requires that the video path be active.

Note: The overvoltage flag for eDP/DP deserializers is VDD_OV_INT_FLAG.

Maxim Confidential Distribute only under NDA
Provided to OEM/ODM Customers Under NDA
ADI Confidential for review under NDA

35.2.2 VDD18 Monitoring Details

All GMSL2 devices include undervoltage monitoring of VDD18.

35.2.2.1 Undervoltage Monitoring of VDD18

An undervoltage condition on VDD18 is nominally flagged when VDD18 < 1.625V. Note that the precise threshold varies some between devices. Refer to the datasheet specific to the part number to verify the specified VDD18 undervoltage threshold for a given device.

In the case of an undervoltage condition, the status register bit CMP_STATUS[0] is latched low. The error can be flagged using the VDDCMP_INT_FLAG, which can be configured to drive the fault condition to the ERRB pin. The error status is cleared by first reading CMP_STATUS[0] and then reading VDDCMP_INT_FLAG.

35.2.3 VDDIO Monitoring Details

VDDIO includes undervoltage monitoring only; overvoltage monitoring is not available. This monitoring is available to all GMSL2 devices. An undervoltage condition on VDDIO is nominally flagged when VDDIO < 1.625V. Note that the precise threshold varies some between devices. Refer to the datasheet specific to the part number to verify the specified VDDIO undervoltage threshold for a given device.

In the case of an undervoltage condition, the status register bit CMP_STATUS[1] is latched low. The error can be flagged using the VDDCMP_INT_FLAG, which can be configured to drive the fault condition to the ERRB pin. The error status is cleared by first reading CMP_STATUS[1] and then reading VDDCMP_INT_FLAG.

35.2.4 VTERM Monitoring Details

VTERM includes undervoltage monitoring only; overvoltage monitoring is not available. An undervoltage condition on VTERM is nominally flagged when VTERM < 1.0V. Note that VTERM is only available in MIPI CSI-2 deserializers.

In the case of an undervoltage condition, the status register bit CMP_VTERM_STATUS is latched low. The error can be flagged using the VDDCMP_INT_FLAG, which can be configured to drive the fault condition to the ERRB pin. The error status is cleared by first reading the PWR0 register and then reading VDDCMP_INT_FLAG.

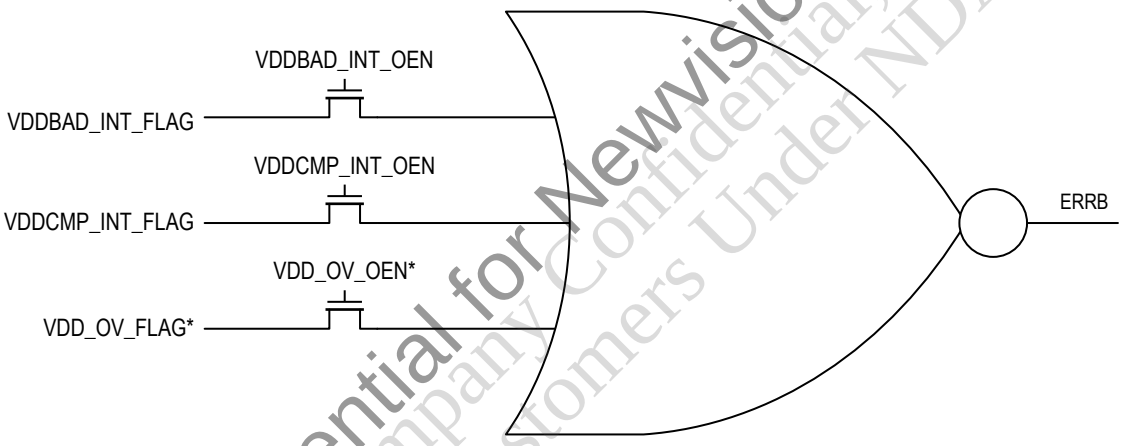
ADI Confidential for Newvision Under NDA
Analog Devices Confidential, Subject to Duty under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

35.3 Error Reporting and Status

35.3.1 ERRB Configuration

The ERRB pin can be used to notify the system of undervoltage and overvoltage conditions. Each of the available monitor functions can be separately routed to the ERRB pin as described in the above sections detailing the various voltage monitors. The user-configurable register fields are listed below. These register fields, when set to 1, enable the reporting of the associated undervoltage/overvoltage condition to the ERRB pins (i.e., the error condition will assert ERRB low). *Figure 133* contains a block diagram of the relationship between the voltage monitoring interrupt enable registers and ERRB pin. Reference the GMSL2 Error Reporting (ERRB Pin) section for additional information.

- VDDBAD_INT_OEN – asserts ERRB low when VDDBAD_INT_FLAG = 1 (VDD_sw UV).
- VDDCMP_INT_OEN – asserts ERRB low when VDDCMP_INT_FLAG = 1 (VDD18, VDDIO, VTERM, and/or VDD_sw UV).
- VDD_OV_OEN – asserts ERRB low when VDD_OV_FLAG = 1 (VDD_sw OV).
    - For eDP/DP deserializers: VDD_OV_INT_OEN – asserts ERRB low when VDD_OV_INT_FLAG = 1 (VDD_sw OV).



*Note: The overvoltage flag for eDP/DP deserializers is VDD_OV_INT_FLAG and the ERRB configuration register is VDD_OV_INT_OEN

Figure 133. Voltage Monitoring ERRB Configuration

35.3.1.1 Latching Status Bits and Clearing Errors

The status flags that drive ERRB are latching. They are set in the event of an error condition; the error indication is persistent following recovery of the error. The flags and other associated bits are cleared automatically when read. Note that some of the flags are driven by status bits that are also latched and must be cleared prior to clearing the flag. Flags associated with overvoltage conditions only become active when the video path is active. If the video is path is not active, the flag bits will not be set in the event of an overvoltage.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

35.4 Feature Availability by Device Family

Refer to device-specific datasheets for feature details and availability. The following table summarizes feature availability by device family (Table 158).

Table 158. Voltage Monitoring Feature Availability by Device Family

Device Family	VDD_sw Undervoltage	VDD_sw Overvoltage	VDD18 Undervoltage	VDDIO Undervoltage	VTERM Undervoltage
HDMI Serializers	X		X	X	
Advanced HDMI Serializers	X		X	X	
DSI Serializers	X	X	X	X	
Advanced DSI Serializers	X	X	X	X	
CSI-2 Serializers	X	X	X	X	
Advanced CSI-2 Serializers	X	X	X	X	
CSI-2 Camera Deserializers	X	X	X	X	X
CSI-2 Display Deserializers	X	X	X	X	X
OLDI Deserializers	X	X	X	X	
eDP/DP Deserializers	X	X	X	X	
CSI-2 Quad Deserializers	X	X	X	X	X

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distributed (Including Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

36 ADC

Section pending.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

37 Line Fault

37.1 Operation

Line-fault detection can be added to GMSL2 systems with the addition of two external resistors at each end of the serial link. Line-fault detection requires that both ends of the cable shield are tied to ground. The external resistor (R_{EXT}) is connected to the LMNx pins (i.e., where the fault condition is to be detected). The external resistor (R_{PD}) is required on the non-detecting side of the link. This scheme detects various application fault conditions, including:

- Short-to-battery
- Short-to-ground
- Open-circuit
- Line-to-line short

The line-fault detection configuration options and status are accessible via registers. If unmasked, a line-fault condition will assert ERRB. Reference the GMSL2 Error Reporting (ERRB Pin) section for additional information.

Note: The external LMN resistor limits the current flowing into the LMN pin to less than 1mA in the event of a short to battery or ground.

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Do not copy or distribute
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

37.2 Hardware Requirements

37.2.1 Coax Mode (Single-Ended)

The local side performs the line fault detection function. The local-side device requires a single 48.7kΩ resistor connected directly from an LMNx pin to the serial link. The remote side of the serial link requires a 49.9kΩ resistor connected to GND, making the remote side hardware-compatible with existing designs using GMSL1 devices. Any of the line-monitor pins may be used when the serial link is single-ended (coax). The line-fault signal assignment is shown in Table 159.

Table 159. Line-Fault Signal Assignment to SIO, and Resistors in Coax Mode

Signal	SIOP (If SIOP is the active PHY)	SION (If SION is the active PHY)
Line Fault Pin	LMN0–3 (any may be used) 48.7kΩ to serial link	LMN0–3 (any may be used) 48.7kΩ to serial link

Note: Line-fault detection can be implemented in either the serializer or deserializer; the orientation is dependent on where the microcontroller is located on the serial link system.

The two configurations for line-fault detection are shown in *Figure 134* and *Figure 135*. Configuration 1 is typically used for display links and configuration 2 is typically used for camera links; however, either configuration can be used in any serial link system.

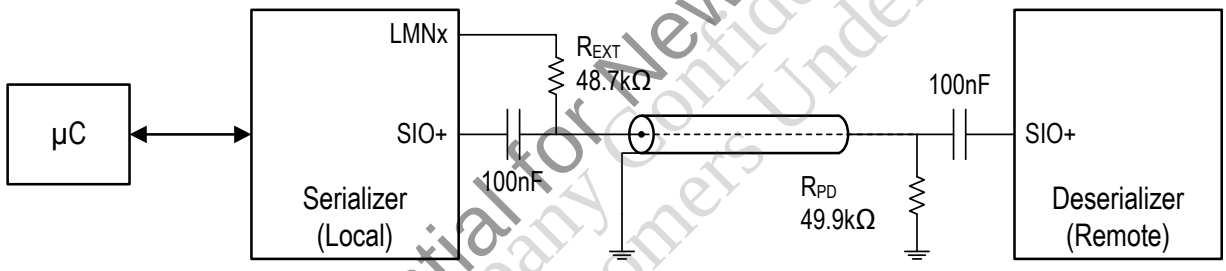


Figure 134. Line-Fault Configuration 1: Local-Side Serializer (Coax Mode)

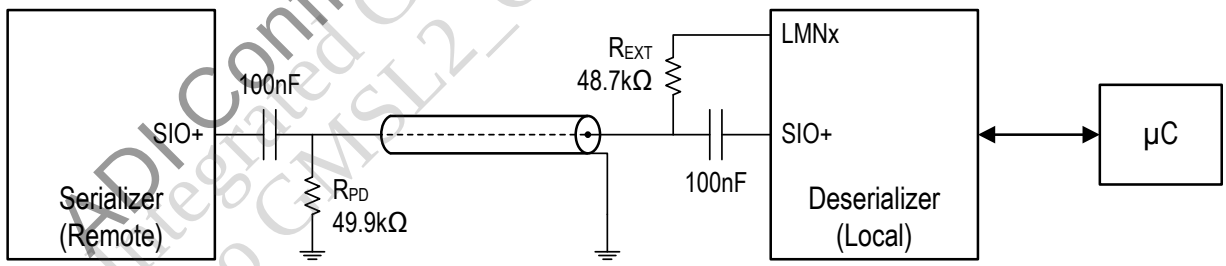


Figure 135. Line-Fault Configuration 2: Local-Side Deserializer (Coax Mode)

The LMNx pins (LMN0 – LMN3) are typically mapped to different multifunctional pins on each unique part and package options. Some parts may have up to four line-fault detectors depending on the package options and pin availability. Refer to device-specific datasheets for more information.

ADI Confidential for New Vision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

37.2.2 Twisted-Pair Mode (Differential)

If the serial link is operating in twisted-pair mode, it is recommended to connect one line to an even-numbered pin and the other line to an odd-numbered pin on the local side. For reliable line-fault detection, the even-numbered pins (i.e., LMN0 and LMN2) must be connected to the line using a 42.2kΩ resistor; the odd-numbered pins (i.e., LMN1 and LMN3) must use a 48.7kΩ resistor to connect to the line (as with single-ended mode). On the remote side, both lines require a 49.9kΩ resistor connected to GND.

For full operation of line-fault detection in twisted-pair applications, ensure that either LMN0/1 or LMN2/3 pairs are used. This pairing is required for proper operation of the line-to-line short detection; however, this pairing is not necessary for the detection of other fault conditions.

Note: These requirements do not apply in coax (single-ended) mode. In coax mode, a 48.7kΩ resistor must be used for any LMNx pin for reliable line-fault detection.

Table 160 and Table 161 show the line-fault detection options for twisted-pair serial link. As shown in the tables, the polarity of the line fault pair is arbitrary with respect to the GMSL signal polarity.

Table 160. Line Fault Pair #1 (LMN0/LMN1) Resistor Values and Connection Choices in STP Mode

Line Fault Pair #1: LMN0 & LMN1	SIOP	SION
Option #1	LMN0: 42.2kΩ to serial link	LMN1: 48.7kΩ to serial link
Option #2	LMN1: 48.7kΩ to serial link	LMN0: 42.2kΩ to serial link

Table 161. Line fault Pair #2 (LMN2/LMN3) Resistor Values and Connection Choices in STP Mode

Line Fault Pair #2: LMN2 & LMN3	SIOP	SION
Option #1	LMN2: 42.2kΩ to serial link	LMN3: 48.7kΩ to serial link
Option #2	LMN3: 48.7kΩ to serial link	LMN2: 42.2kΩ to serial link

A typical hardware setup for twisted-pair line monitoring is shown in Figure 136.

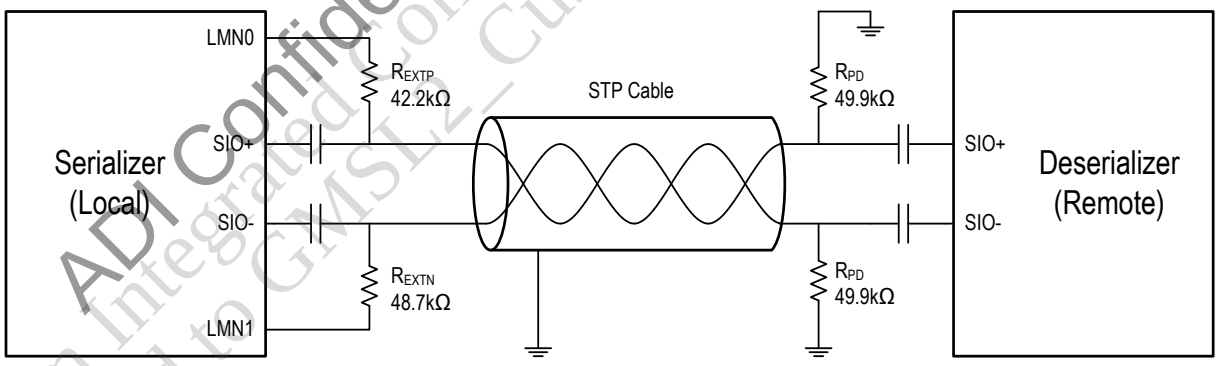


Figure 136. Line-Fault Detection Configuration: Local-Side Serializer (Twisted-Pair Mode)

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential
Provided to GMSL2_ Customers Under NDA

37.3 Configuration

GMSL2 device register control allows access to enable the line-fault detectors, read the line-fault status codes, and program the line-fault interrupts. The line-fault registers are outlined below in Table 162.

Table 162. Register Mapping and Descriptions for the Line-Fault Registers

Register Name	Bit(s)	Bit Name	Bit Description
REG5	3	PU_LF3	Power up Line-Fault Detector 3 (LMN3) if applicable
REG5	2	PU_LF2	Power up Line-Fault Detector 2 (LMN2) if applicable
REG5	1	PU_LF1	Power up Line-Fault Detector 1 (LMN1) if applicable
REG5	0	PU_LF0	Power up Line-Fault Detector 0 (LMN0) if applicable
REG27	6:4	LF_3[2:0]	LMN3 status (see *Table 163* Decodes)
REG27	3:0	LF_2[2:0]	LMN2 status
REG26	6:4	LF_1[2:0]	LMN1 status
REG26	3:0	LF_0[2:0]	LMN0 status
INTR2	3	LFLT_INT_OEN	Sends line-fault interrupt to ERRB pin
INTR3	3	LFLT_INT	Line-fault interrupt asserted when any of the four enabled detectors indicates a fault condition

Note: The behavior of the line-fault interrupt can be selected to be either latching or live with the LFLT_STKY_INT bit (interrupt is live by default). Feature availability varies by device; refer to device-specific datasheet for availability details.

The line-fault detection status registers are encoded to represent various conditions. The code translations are presented below in Table 163. Unused detectors (PU_LFx = 0) will return “010” (i.e., the status code for ‘Normal Operation’).

Table 163. Line-Fault Detection Decode Table

Line-Fault Condition	LF_0,1,2,3 [2:0]
Short-to-battery	000
Short-to-GND	001
Normal Operation (no fault)	010
Open-line	011
Line-to-line short	1xx

37.3.1 Line-Fault Detection Application

To use line-fault detection, power up the line-fault detector (REG5) that corresponds with the pin used on the device that will be reporting the error condition (local device). For example, if using pins LMN0 and LMN1, set the local-side line-fault REG5 register bits PU_LF0 and PU_LF1 to 1. Do not enable the line-fault detector on the remote side.

In normal operation, the status registers LF_0 [2:0] and LF_1 [2:0] will each return “010” when read. If a fault is detected, the status registers will change according to error condition (see Table 163). Unused line-fault detectors will default to the “010” status code.

ADI Confidential for Review Only
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

For example, assume LMN0 is connected to SIO+ through R_{EXT} as shown in *Figure 134*. If the serial link (SIO+) is pulled to GND, LF_0[2:0] will return a code of "001", and the interrupt bit LFLT_INT in the INTR3 register will be set to 1. If LFLT_INT_OEN in the INTR2 register is enabled, the ERRB pin will transition low to reflect the line-fault interrupt condition.

37.3.1.1 Operation During Line-Fault Detection Events

The SIO+/SIO- pins are protected by AC coupling capacitors and are not exposed to high voltage caused by a line-fault event. In the event of a short-to-battery, the current into the LMN pin is limited to less than 1mA by the external LMN resistor. Following a line-fault event, no immediate action is required, however, system designers should take the appropriate steps to verify the system condition. ERRB will remain low until the line-fault event is reversed.

Note: If ERRB is low, follow the procedure described in the *Example ERRB System Reaction* section to determine the source of the error condition.

37.3.1.2 Line Fault with PoC Recommendation

Line-fault monitoring cannot be enabled while using PoC. ADI recommends a supervisory power supply device for line fault detection and monitoring PoC voltage and current. The MAX20086–MAX20089 devices are designed specifically to work with all GMSL devices in automotive camera applications and are ASIL-compliant for safety-critical functions (ASIL-B and ASIL-D compliant versions are available). Reference the ASIL section for more information.

37.4 Feature Availability by Device Family

Line-fault detection can be configured for all GMSL2 devices.

ADI Confidential for New Version under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

38 Error Generator (ERRG)

All GMSL2 devices have an error generator (ERRG) located after the packet scheduler to simulate the effect(s) of bit errors on the link. This is primarily used to test system-level reactions to serial link bit errors, including ASIL error handling in safety-relevant systems. The ERRG can also be used to validate internal self-tests (e.g., PRBS Testing).

38.1 Operation

The error generator is located within each PHY after the scheduler and packetizer. When enabled in the serializer, bit errors are added to the forward channel; when enabled in the deserializer, bit errors are added to the reverse channel. Bit errors are generated by flipping bits in the bitstream after data packetization and encoding, so there is no control over what channels (e.g., video, RMII, etc.) are affected.

38.2 Configuration

To enable the ERRG, first set the ERRG parameters in register TX2 (Table 164). Then, enable the error generator in register TX1 (Table 165) for all devices except CSI-2 Quad Deserializers, which use register GMSL__x:TX1 (Table 166).

Note: To clear ERRB status after disabling ERRG mode (i.e., ERRG_EN_A or ERRG_EN_B = 0), clear as described in the *GMSL2 Error Reporting (ERRB Pin)* section or perform a one-shot reset.

38.2.1 ERRG Parameters

The following parameters must be configured in register TX2 before enabling the error generator:

- ERRG_RATE[1:0] – The error generation rate controls how often error events are triggered (i.e., the generated BER). The recommended setting is ERRG_RATE[1:0] = 11. This sets a BER of 4.77 * 10^{-8}, which is four orders of magnitude worse than the expected worst-case BER for a compliant GMSL2 link.

- ERRG_CNT[1:0] – The error count determines how long to run the ERRG. In continuous mode, the ERRG will produce errors as long as ERRG_EN_x is set high. For the other values of ERRG_CNT, the ERRG will generate a specified number of error events at the rate specified by ERRG_RATE[1:0]. Setting ERRG_EN_x low will reset the error count.

- ERRG_PER[0] – Sets the error generation mode. In periodic mode, the ERRG generates a bit flip at exactly the rate specified by ERRG_RATE[1:0]. In pseudorandom mode, the bit flip is generated at a random point by an internal pseudorandom number generator at a rate defined by ERRG_RATE[1:0] so that the average rate of error generation is equal to the rate specified.

- ERRG_BURST[2:0] – Error burst defines how many errors are generated per error event. By default, this is set to '000' and generates a single error per error event. If set to a value larger than 1 bit, a burst of sequential errors is generated: the first and last bits are flipped, and the other bits have a 50% probability of being flipped. For example, if a value of 8 bits is selected, the first and eighth bits will be flipped, and bits 2–7 will each have a 50% chance of being flipped.

Analog Confidential Information only
Maxim Integrated GMSL2 Confidential
Provided to GMSL2 Customers Under NDA
Analog Confidential for New vision under NDA

Table 164. ERRG Parameters (Register TX2)

Parameter	Bitfield	Decode
Select the error generation mode	ERRG_PER[0]	0: Pseudorandom 1: Periodic
Set the burst length of errors to be generated at every error event	ERRG_BURST[2:0]	000: 1 bit 001: 2 bits 010: 3 bits 011: 4 bits 100: 8 bits 101: 12 bits 110: 16 bits 111: 20 bits
Set error event generation rate	ERRG_RATE[1:0]	00: 1 error in 5120 bits 01: 1 error in 81,920 bits 10: 1 error in 1,310,720 bits 11: 1 error in 20,971,520 bits
Set the number of error events to be generated after ERRG_EN is set high.	ERRG_CNT[1:0]	00: Continuous 01: 16 10: 128 11: 1024

38.2.1.1 ERRG Enable: All Device Families Except CSI-2 Quad Deserializers

Table 165. ERRG Enable (Register TX1)

Parameter	Bitfield	Decode
Enable the error generator for the desired GMSL PHY	ERRG_EN_A ERRG_EN_B	0: Disabled 1: Enabled

38.2.1.2 ERRG Enable: CSI-2 Quad Deserializers

Table 166. ERRG Enable (Register GMSL__x:TX1)

Parameter	Bitfield	Decode
Enable the error generator for the desired GMSL PHY	ERRG_EN	0: Disabled 1: Enabled

Note: There is a GMSL__x:TX1 register associated with each of the four GMSL PHYs.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Under NDA
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2 Customers

38.3 Availability by Device Family

The error generator function is available to all GMSL2 devices (Table 167).

Table 167. Availability by Device Family

Device Family	ERRG
HDMI Serializers	X
Advanced HDMI Serializers	X
DSI Serializers	X
Advanced DSI Serializers	X
CSI-2 Serializers	X
Advanced CSI-2 Serializers	X
CSI-2 Camera Deserializers	X
CSI-2 Display Deserializers	X
OLDI Deserializers	X
eDP/DP Deserializers	X
CSI-2 Quad Deserializers	X

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Limited to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Device Families

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

39 HDMI Serializers

39.1 Overview

All GMSL2 HDMI serializers convert audio and video from HDMI 1.4 or 2.0 input to single or dual GMSL2 output. The HDMI receiver (HDMI Rx) block provides HDCP 1.4 or HDCP 2.3 decryption of the received HDMI stream; select HDMI serializers can also encrypt data transmitted across the serial link (provided the connected deserializer has HDCP decryption capability). Refer to the HDCP section for more details.

GMSL2 HDMI serializers support a maximum HDMI video pixel clock rate of 422MHz; advanced HDMI serializers support a video pixel clock rate up to 600MHz with compression (the maximum speed defined in HDMI 2.0 Specification). The maximum active horizontal resolution is 5760 when using compression. Display Stream Compression (DSC) is required if the input PCLK exceeds the GMSL bandwidth capacity (i.e., 215MHz PCLK per link). Refer to the Display Stream Compression (DSC) section for more details.

See Feature Availability by Device for a list of GMSL2 HDMI devices.

A functional block diagram below (Figure 137) shows the video path from the HDMI input pins to the video pipes.

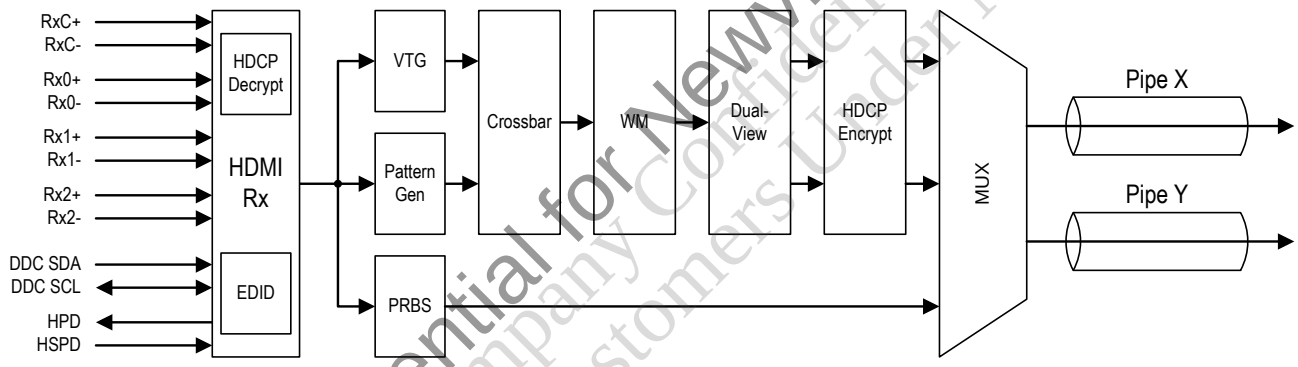


Figure 137. Block Diagram of the Video Path Through GMSL2 HDMI Serializers

39.2 Operation

The HDMI receiver in the GMSL2 HDMI serializers supports RGB888 format video input. The HDMI receiver includes the following functions/features:

- HDMI 1.4 and 2.0 input.
- HDMI audio receiver and transmission over the forward channel.
- HDCP 1.4 and 2.3 decryption.

Analog Devices Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

39.2.1 Use Cases: Embedded and Consumer Port Applications

39.2.1.1 Embedded Applications

GMSL2 HDMI serializers are most commonly used in embedded applications where the video source (typically an SoC) is installed on the same PCB as the GMSL2 serializer (*Figure 138*). In these implementations, a PCB trace provides the HDMI connection from the video source to the serializer; the subsystem is contained within the PCB, and there are no external off-board connections. This use case requires careful consideration of PCB design and layout to ensure good signal quality on the HDMI connection. HDMI compliance testing is not required for this application.

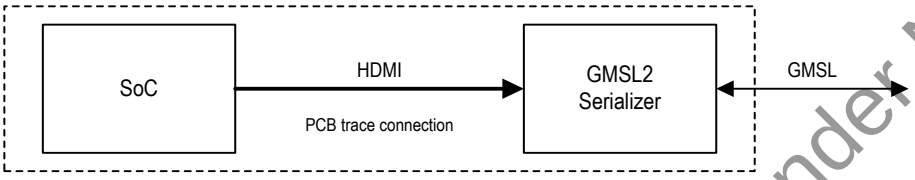


Figure 138. Embedded HMDI Application

39.2.1.2 Consumer Port Applications

HDMI consumer port applications integrate an off-board HDMI connection (Figure 139) that allows end users to connect an external video source. Rear seat infotainment systems with an external HDMI connector are a common example of consumer port applications. This use case typically requires HDMI compliance certifications. To be compliant, the PCB layout must meet HDMI hardware requirements, and the serializer must be configured according to the register optimization process (see HDMI Receiver Optimization Configuration).

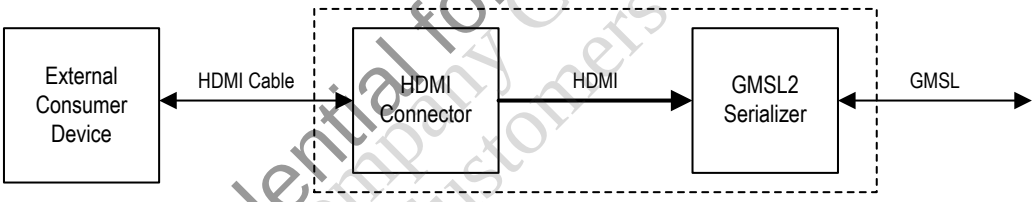


Figure 139. Consumer Port Application

ADI Confidential for New Version under NDA
Analog Devices Confidential Distributed Only to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

39.3 Configuration

The following configuration steps are used to enable video in the HDMI serializers and advanced HDMI serializers.

- Configure the EDID table (if required). See HDMI EDID Tables for details.
- Set the HDMI mode. See the HDMI Mode (HDMI 1.4 or HDMI 2.0) section below for configuration details.
- For consumer port applications, write HDMI receiver optimization registers (HDMI Receiver Optimization Configuration).
- Set up dual-view registers (registers DV0 and DV2). These are disabled by default. See the Dual-View section for additional details.
- Set the stream ID for video. The TX_STR_SEL bits set the stream ID mapping for each video pipe.
    a. TX3 (0x0053): Controls pipe X (Default stream ID is 0). Selects which stream ID to transmit the main HDMI video stream.
    b. TX3 (0x0057): Controls pipe Y (used when dual-view is enabled). See the Video Pipes section for additional information.
- Enable pixel repetition mode (if required). Pixel repetition mode must be used for PCLK values under 25MHz. See the HDMI Pixel Repetition Mode section for additional information.
- Configure HDMI audio (if required).
- Configure HDCP (if required). See the HDCP section for additional information.
- Enable HDMI input block by setting HDMI_AUTOS = 1 (register REG1). This is required to receive HDMI video.

39.3.1 HDMI EDID Tables

Extended Display Identification Data (EDID) is a data structure used to describe the capabilities of a display (i.e., sink device) to a video source with the VESA defined standard format. An EDID table loaded in an HDMI sink device communicates the sink device's supported resolutions, timing information, manufacturer ID, serial number, etc. to the HDMI source device(s). GMSL2 serializers can store EDID data and share it with the source device (Figure 140).

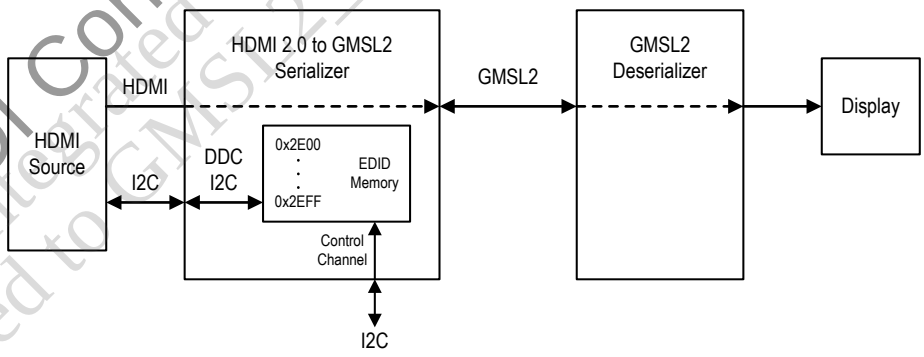


Figure 140. EDID Memory in the Serializer HDMI Rx Block

ADI Confidential for Newvision under NDA
Analog Confidential Distribution
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

A typical application of EDID tables is to ensure functional compatibility of computer monitors and graphic cards. The EDID table in the computer monitor is read by the computer's graphics card so that the necessary video characteristics are generated and output on the HDMI port.

39.3.1.1 EDID Table Usage Requirement

An EDID table may be required depending on the HDMI source used in the system. For consumer port applications or when using a computer as a video source, an EDID table must be loaded to ensure proper operation. For embedded applications, however, an EDID table is typically not required as the video resolution is usually hard coded in the SoC and EDID tables are ignored.

Note: For consumer port applications or when using a computer as a video source, the HDMI source device must first see a valid EDID table in the HDMI sink device (i.e., the HDMI serializer) for it to recognize the presence of an HDMI sink device before it will output video.

39.3.1.2 Sink Device EDID Memory

The GMSL2 HDMI serializer acts as the sink device and functionally represents the display to the HDMI source. The GMSL2 HDMI serializer provides 256 registers for EDID storage (accommodating EDID structure v2.0). The EDID data is written to volatile memory space in the serializer through the device's primary I2C or UART interface. The EDID register block is 0x2E00–0x2EFF.

Note: EDID data must be written to the serializer whenever the device is power cycled.

39.3.1.3 Source EDID Access

The HDMI source must access the EDID data through the Display Data Channel (DDC) at the HDMI interface. The DDC is an I2C interface and is separate from the serializer's I2C bus.

39.3.1.4 EDID Registers

The EDID table memory space is located in registers 0x2E00–0x2EFF. When the HDMI source device in the HDMI link detects HPD high (hot plug detect), it reads the EDID table through the DDC_SDA and DDC_SCL lines of the HDMI interface.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

39.3.1.5 EDID Configuration

If an EDID table is needed, the typical configuration steps are:

1. Write register to drive HPD pin low (register 0x20F5 set to 0x00).
2. Write EDID table (registers 0x2E00–0x2EFF).
3. Write register to drive HPD pin high (register 0x20F5 set to 0x01).

By toggling the HPD pin, the source recognizes that there is a new EDID table to read. If HPD is not cycled low then high, the source device may not re-read the new EDID table.

Note: HDMI source devices do not use a universal method of polling the status of the HPD pin. Some HDMI source devices detect an edge change on the pin and immediately react; some devices poll the pin level at a set time interval. For certain devices, this polling interval can be up to multiple seconds. Ensure the delay between steps 1 and 3 (above) is sufficient for the target source device to recognize the HPD level change.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

39.3.2 HDMI Mode (HDMI 1.4 or HDMI 2.0)

HDMI serializers receive video input from an HDMI source via the TMDS (Transition-Minimized Differential Signaling) interface. TMDS is an 8b/10b encoding scheme with an interface consisting of a differential set of clock signals and three sets of differential data channel signals.

In HDMI 1.4 mode, the three sets of data channels operate at 10x of the pixel clock frequency (up to 3.4Gbps). The TMDS clock signal pair has the same frequency as the pixel clock. There is no phase requirement between the clock signals and the data channels. In HDMI 2.0 mode, the frequency of the TMDS clock signal pair is divided by four. The data channels' rate is 40x the TMDS clock rate. Data scrambling is required to reduce EMI/RFI.

- HDMI 1.4 mode (default) is used for PCLK values under 340MHz.
- HDMI 2.0 mode is used for PCLK values between 340MHz and 600MHz.

39.3.2.1 HDMI 1.4 Mode Configuration

By default, the HMDI Rx block is configured for HDMI 1.4; this mode **MUST** be used if the source PCLK is 340MHz or below. No configuration is needed to enable the HDMI input block in HDMI 1.4 mode.

39.3.2.2 HDMI 2.0 Mode Configuration

HDMI source devices utilize HDMI 2.0 to transmit videos with a PCLK value above 340MHz. HDMI serializers **MUST** be configured to HDMI 2.0 mode to receive this input. Enable HDMI 2.0 mode using the following register writes:

```
# Enable HDMI 2.0 Mode
# Hold HDMI receiver digital logic in reset
0x80,0x2005,0x01
# Enable all 3 TMDS data channels
0x80,0x3D0C,0x20
0x80,0x3E8D,0x40
# HDMI2 overwrite, scdt on
0x80,0x3E50,0x23
# 5MHz OSC for Zone detect reset
0x80,0x3C04,0x04
# Release reset of 5MHz OSC
0x80,0x3C04,0x00
0x80,0x230E,0x24
0x80,0x230F,0x00
0x80,0x233B,0x80
0x80,0x2301,0x00
# Scramble-on overwrite, HDMI2-on overwrite
0x80,0x2040,0x33
0x80,0x2313,0x20
# Release HDMI receiver digital reset
0x80,0x2005,0x00
```

Maxim Confidential
Provided to GMSL2_Customers Under NDA
Maxim Integrated Company Confidential, Distribution only under NDA
Public Confidential for New Vision under NDA

39.3.2.3 Status and Control Data Channel (SCDC) for TMDS Configuration

In accordance with the HDMI 2.0 specification, the serializer contains the SCDC register for TMDS configuration. For HDMI 2.0 consumer port applications, the EDID programmed into the serializer should contain the HF-VSDB (HDMI-Forum Vendor-Specific Data Block) SCDC_Present bit set high to indicate that it has implemented the SCDC.

Note: The EDID must be correctly programmed for HDMI 2.0 operation. The HF-VSDB block in the EDID indicates that the sink device supports HDMI 2.0 features (HDMI 2.0 specification §10.3.2). If an HDMI source does not find this block in the HDMI sink device, the source will assume it is connected to an HDMI 1.4 device.

39.3.3 HDMI Receiver Optimization Configuration

For consumer port applications, the following configuration should be used to optimize the HDMI receiver equalizer for jitter tolerance performance needed to pass HDMI compliance tests. The receiver optimization configuration is needed for HDMI 1.4 mode; no optimization is required for HDMI 2.0 mode.

39.3.3.1 HDMI 1.4 Receiver Optimization

```
# Script to set up HDMI 1.4 RX EQ Optimizations
0x80,0x3D0A,0x1F
0x80,0x3E5C,0x0D
# CTS EQ Lookup table - PEQ_VAL0-7
0x80,0x3E08,0x18
0x80,0x3E09,0x38
0x80,0x3E0A,0x58
0x80,0x3E0B,0x78
0x80,0x3E0C,0x98
0x80,0x3E0D,0xB8
0x80,0x3E0E,0xD8
0x80,0x3E0F,0x78
# cdr bypass
0x80,0x3D0E,0x82
# bp_fix
0x80,0x3E15,0x40
# auto mode and auto engine decision enable
0x80,0x3E00,0x20
```

Note: This configuration is not compatible with HDMI 2.0 operation.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute Worldwide
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

39.3.3.2 HDMI 2.0 Receiver Operation

For consumer port applications in HDMI 2.0 mode, no optimization is needed. For applications that switch between HDMI 1.4 mode and 2.0 mode, the script below should be used to disable the HDMI 1.4 receiver optimizations.

```
# Restore 2.0 EQ Optimizations with default values
0x80,0x3D0A,0x18
0x80,0x3E5C,0x1D
# Restore cdr_ctl1 default value
0x80,0x3D0E,0x02
# Revert bp_fix to default value
0x80,0x3E15,0x60
# Restore auto engine decision enable (reg_cfg_reg_edon) to default
0x80,0x3E00,0x00
```

39.3.3.3 Dual Mode Applications (HDMI 1.4 and HDMI 2.0 Modes)

For consumer port applications that support multiple video resolutions, additional steps are required to detect the HDMI mode sent by the video source. In compliance with the HDMI specification, the HDMI source uses the DDC channel to set registers in the HDMI receiver in the serializer. These registers are located in the serializer register block DDC_SCDC_REG (address 0xA801 to 0xA8FF). For HDMI2.0 operation, the source should program reg_scrbl_enable (Scramble Enable) high and reg_tmds_bclk_ratio (TMDS bit clock ratio) high in register DDC_SCDC_REG TDMS_Config (register 0xA820). A change in value of the bit reg_scbl_enable can be configured to trigger an interrupt to the ERRB pin. The interrupt event can then be used to inform the application controller that the HDMI source has changed modes and can reconfigure the system (including the HDMI serializer) into the correct mode.

This interrupt sequence requires that the register reg_intr7_stat6_aon (register 0x2082) be enabled and interrupt reg_intr7_mask6_aon (register 0x2092) be enabled. In addition, the HDMI_INT_OEN (register 0x001A) and reg_rxdig_intr_mask (0x3C11) registers must be enabled. Refer to the HDMI Video Status and Interrupt section for additional details.

ADI Confidential for New Non-disclosure under NDA
Analog Devices Confidential Information Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

39.3.4 HDMI Pixel Repetition Mode

HDMI video sources use a pixel repetition (replication) scheme when the selected video format produces a pixel clock slower than 25MHz. This scheme duplicates pixels and increases the pixel clock frequency. GMSL2 HMDI serializers can be programmed to remove the replicated pixels in multiples of two, four, or eight.

39.3.4.1 Pixel Repetition Removal

GMSL2 HDMI serializers have a pixel repetition mode that can return the received video to the original video format and rate if the HDMI source device uses the pixel repetition scheme. Refer to Table 168 for pixel repetition mode details.

Table 168. Pixel Repetition Removal

Register Address	Bitfield	Description	Decode
0x2A15	Pixel_Rate[1:0]	Remove replicated pixels	0x0: Pixel Replication 1x 0x1: Pixel Replication 2x 0x2: Pixel Replication 4x 0x3: Pixel Replication 8x

39.3.4.2 Pixel Repetition Removal Test

Pixel repetition removal can be tested by enabling an internal test mode and observing the pixel clock. To enable the internal test mode and verify pixel repetition mode, program the device according to the procedure below:

- Write the following sequence using low-level commands:

```
0x80,0x003F,0xA0
0x80,0x003F,0xC0
0x80,0x003F,0xCF
0x80,0x003F,0xA0
0x80,0x003F,0x06
0x80,0x003E,0xC0
```

- Observe the pixel clock on GPIO13 (SDOR/ADD0).
- Set the signal source to 1360x768(60). The observed pixel clock at GPIO13 should be 85MHz.
  - If preferred, an alternate timing can be used. The expected clock from the specified timing should be observed at GPIO13.
- Change the signal source to 480i (PCLK = 13.5MHz). The measured pixel clock should be 27MHz due to pixel repetition at the source.
- Write register 0x2A15 to value 0x1 for pixel repetition of 2x. The measured pixel clock will change to 13.5MHz (the original PCLK rate) if pixel repetition mode is operating correctly.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL 2 Customers Under NDA

39.4 HDMI Rx Errors and Debug

This section explains how to detect if a GMSL2 HDMI serializer is receiving video and how to read out the detected pixel clock and video timing parameters (i.e., resolution and blanking).

39.4.1 HDMI Serializer Video Detection

GMSL2 HDMI serializers contain debug registers that can be used to detect if the device is receiving a valid input video signal (Table 169).

Table 169. GMSL2 HDMI Serializer Debug Registers

Register	Register Address	Bit name (bit)	Description
VIDEO_TX2	0x0102	PCLKDET (bit 7)	High if a stable PCLK is detected on HDMI input (Serializer PLL locked to PCLK)
VIDEO_TX2	0x0102	DRIFT_ERR (bit 6)	High if the PCLK has drifted significantly
VIDEO_TX2	0x0102	OVERFLOW (bit 5)	High if video transmit buffer in serializer has overflowed
VIDEO_TX2	0x0102	FIFO_WARN (bit 4)	High if the video FIFO is more than half full
VTX1	0x01C9	HDMI_CKDT (bit 6)	High if a HDMI clock is detected
VTX1	0x01C9	HDMI_SCDT (bit 7)	High if a HDMI sync is detected

39.4.2 Received Video Timing

GMSL2 HDMI serializers have a register block that is used to determine the received PCLK and video timing parameters. Each parameter is reported as a hexadecimal number output across two registers, LSBs in the first and MSBs in the second register. Values are reported less than two seconds after HDMI_AUTOS = 1.

The registers that report the video timing parameters are listed below.

- Horizontal active resolution = value(0x2A8C) + (value(0x2A8D)*0x100)
- Vertical active resolution = value(0x2A8E) + (value(0x2A8F)*0x100)
- HS low period = value(0x2A90) + (value(0x2A91)*0x100)
- HS high period = value(0x2A92) + (value(0x2A93)*0x100)
- Horizontal front porch = value(0x2A94) + (value(0x2A95)*0x100)
- Horizontal back porch = value(0x2A96) + (value(0x2A97)*0x100)
- VS low period = value(0x2A98) + (value(0x2A99)*0x100)
- VS high period = value(0x2A9A) + (value(0x2A9B)*0x100)
- Vertical front porch = value(0x2A9C) + (value(0x2A9D)*0x100)
- Vertical back porch = value(0x2A9E) + (value(0x2A9F)*0x100)

Note: ‘value(0xXXXX)’ refers to a read to address 0xXXXX in the HDMI serializer.

API Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

39.4.3 Pixel Clock Parameters

These parameters are reported by the PCLK PLL. This reported PCLK value is an estimate with the following accuracies:

- ±10% accurate between 50MHz–300MHz.
- ±5% accurate above 300MHz.

Values reported in hexadecimal:

- ODIV = VID_ODIV_L[7:0] + (VID_ODIV_H*0x100)
- FBDIV = FB_DIV[5:0]
- FBDIVFRAC = FB_DIV_FRAC[5:0]

To calculate PCLK, convert hex values to decimal:

- PCLK (MHz) = $\frac{75}{ODIV} * \left(FBDIV + \frac{FBDIVFRAC}{64}\right)$
- Total Pixels = $(HS\ low\ period + HS\ high\ period) * (VS\ low\ period + VS\ high\ period)$
- Frame rate = $\frac{PCLK * 10^6}{Total\ Pixels}$

Note: The received PCLK and video timing parameters can also be shown in the GUI. With the HDMI serializer connected to the GUI, the relevant dashboard is located in Tools -> Configure Display Ser Des -> HDMI Timing.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

39.4.4 HDMI Video Status and Interrupts

HDMI event status registers and interrupts are used for a variety of purposes depending on device implementation and system application. This section describes and details the configuration of the status registers, interrupt registers, and masks.

39.4.4.1 Video Status Registers

Video status registers provide information about the presence of the pixel clock, HDMI clock, and sync character. These three detect signals are present in the VTX1 (0x1C9) register. These status bits are read-only and do not drive the ERRB pin. Since HDMI data is spread across the three HDMI TMDS (Transition-Minimized Differential Signaling) signals, valid sync character detection gives confidence in the validity of the incoming video stream.

Register VTX1 (0x01C9):

- Bit 5: PCLKDET – PCLK Detect
- Bit 6: HDMI_CKDT – HDMI Clock Detect
- Bit 7: HDMI_SCDT – HDMI Sync Character Detect

39.4.4.2 HDMI Interrupts

HDMI serializers can detect certain HDMI events and create attendant interrupt signals. These interrupt signals, readable in status registers, can be further configured to output to the ERRB pin to create a hardware interrupt, as shown in Figure 141.

The following interrupts are available:

- **HDMI Sync Detect** – indicates sync characters are stable and detected across all three HDMI TMDS lanes.
- **HDMI Clock Detect** – indicates HDMI clock is detected.
- **HDMI Audio Sample Frequency Change** – indicates change in audio sample frequency (Fs).
- **HDMI Cable Unplug** – indicates HDMI HSPD pin has been disconnected from HDMI source.
- **HDMI scramble_en Change** – indicates HDMI source change from HDMI 1.4 to HDMI 2.0 mode.
- **HDMI Cable Plug-in** – indicates HDMI HSPD pin has been pulled high from HDMI source.

39.4.4.3 Interrupt Status

Each interrupt has a latched status register (Table 170). Write 0x1 to clear the bit.

Table 170. HDMI Rx Interrupt Status Registers

Register Address	Register Name	Bitfield	Decode
0x2080	RX_INTR2_AON	reg_intr2_stat3_aon	Bit 3: HDMI Sync Detect change Interrupt
0x2080	RX_INTR2_AON	reg_intr2_stat4_aon	Bit 4: TMDS Clock Detect change Interrupt
0x2344	RX_INTR5	reg_intr5_stat0	Bit 0: Audio sample frequency change event interrupt
0x2081	RX_INTR6_AON	reg_intr6_stat0_aon	Bit 0: HDMI Cable Unplug Interrupt
0x2082	RX_INTR7_AON	reg_intr7_stat6_aon	Bit 6: HDMI TMDS register scramble_en change event interrupt (indicates a change between HDMI 1.4 and HDMI 2.0 mode)
0x2083	RX_INTR8_AON	reg_intr8_stat1_aon	Bit 1: HDMI Cable Plug-in Interrupt

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA

39.4.4.4 Interrupt Configuration

To enable an interrupt to be output to the ERRB pin:

- Set the interrupt enable bit given in Table 171.
- Enable HDMI receiver "Rx digital interrupt" by setting reg_rxdig_intr_mask to 1 (0x3C11).
- Set HDMI_INT_OEN (0x001A) to 1.

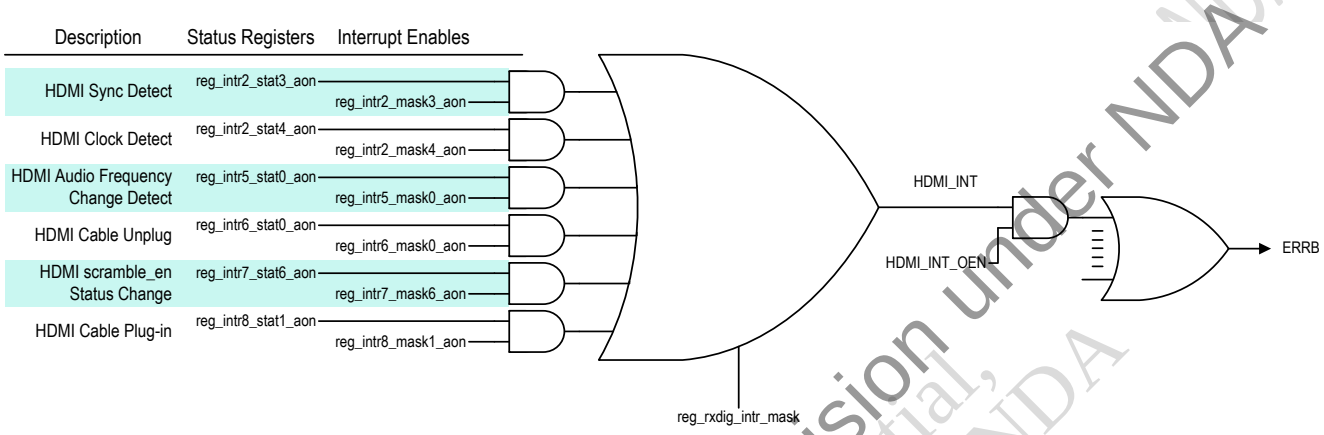


Figure 141. HDMI Fault/Error Register Architecture

Table 171. GMSL2 HDMI Rx Interrupt Enable

Register Address	Register Name	Bitfield	Decode
0x2090	RX_INTR2_MASK_AON	reg_intr2_mask3_aon	Bit 3: HDMI Sync Detect change Interrupt Enable
0x2090	RX_INTR2_MASK_AON	reg_intr2_mask4_aon	Bit 4: TMDS Clock Detect change Interrupt Enable
0x2354	RX_INTR5_MASK	reg_intr5_mask0	Bit 0: Enable audio sample frequency change event interrupt
0x2091	RX_INTR6_MASK_AON	reg_intr6_mask0_aon	Bit 0: HDMI Cable Unplug Interrupt Enable
0x2092	TX_INTR7_MASK_AON	reg_intr7_mask6_aon	Bit 6: Enable Interrupt for scramble_en change event (indicates a change between HDMI 1.4 and HDMI 2.0 mode)
0x2093	RX_INTR8_MASK_AON	reg_intr8_mask1_aon	Bit 1: HDMI Cable Plug-in Interrupt Enable

The GMSL2 HDMI serializer's receiver error bit, which is one of several sources of the ERRB interrupt (refer to GMSL2 Error Reporting (ERRB Pin) section), is status bit HDMI_INT (register INTR3, address 0x001B). To clear the HDMI_INT status, clear the interrupts enabled in registers 0x2080, 0x2081, 0x2083 by writing each status bitfield high (Table 170). Reference Table 172 for the HDMI serializer interrupt enable registers.

ADI Confidential for NewVision under NDA
Maxim Integrated Company Confidential, Distributed to GMSL2 Customers Under NDA
Provided to GMSL2 Customers Under NDA

Table 172. GMSL2 HDMI Serializer Interrupt Enable Registers

Register Address	Register Name	Bitfield	Decode
0x3C11	INTR_STAT0	reg_rxdig_intr_mask	Bit 7: HDMI Error Enable (Direct HDMI Interrupts to serializer error handling)
0x001B	INTR3	HDMI_INT	Bit 4: HDMI_INT status
0x001A	INTR2	HDMI_INT_OEN	Bit 4: Enables reporting of HDMI interrupt (HMDI_INT) to the ERRB pin

39.5 Feature Availability by Device

HDMI is only available on the HDMI and Advance HDMI serializer device families (Table 173).

Table 173. HDMI Device Family Part Numbers

Device Family	Part Numbers	Max PCLK/Link	DSC
HDMI Serializers	MAX96751	215MHz	
	MAX96751F	160MHz	
	MAX96753	215MHz	
	MAX96753F	160MHz	
	MAX96751H	104MHz	
Advanced HDMI Serializers	MAX96763	600MHz	X
	MAX96765	600MHz	X
	MAX96759	600MHz	X
	MAX96761	600MHz	X
	MAX96785	600MHz	X
	MAX96787	600MHz	X

ADI Confidential for New vision under NDA
Analog Devices Confidential, Copyright
Maxim Integrated Company Confidential,
Provided to GMSL2_ Customers Under NDA

40 DSI Serializers

40.1 Overview

GMSL2 DSI (Display Serial Interface) serializers convert MIPI DSI input(s) to GMSL2 or GMSL1 output(s). The MIPI D-PHY DSI input ports have configurable lane arrangements that allow ganged mode configurations. These ports operate independently and offer different video timings of video stream(s) at each port. RGB888 (24-bit) color is the only supported color format.

The required register programming and example use cases are provided in the *Configuration* section.

GMSL2 DSI serializers include the following features and functions:

- Video Crossbar
- GMSL1 backward compatibility
- Dual MIPI D-PHY v1.2 Interface*
- Dual DSI v1.3 Rx Controllers*
- Video duplication*
- Symmetric Dual-View*
- Asymmetric Dual-View*
- HDCP 1.4*
- ASIL-B compliant (GMSL2 mode) *

* Availability dependent on device variant. Refer to *Feature Availability by Device* for more information.

40.2 Operation

GMSL2 DSI serializers convert dual independent MIPI D-PHY v1.2 four-lane input to single, dual, or mixed GMSL1 or GMSL2 serial link output. The MIPI D-PHY receiver block converts RGB888 DSI packets to PPI (PHY Protocol Interface). The PPI packets are then converted to DPI parallel video data with horizontal sync (HS) and vertical sync (VS) timing information by the DSI receiver controller. The GMSL transmit block serializes and transmits this data to connected deserializer device(s) over the serial link.

The GMSL2 DSI serializer may have up to two MIPI receiver ports (depending on device package). Each port comprises one or two MIPI D-PHY v1.2 receiver(s) (depending on device configuration) and is connected to one MIPI DSI v1.3 Rx controller. The MIPI port consists of a single differential clock lane and up to four differential data lanes with a maximum data rate of 2.5Gbps per lane. Each receiver provides up to 10Gbps total bandwidth shared across four data lanes. Refer to *Feature Availability by Device* for more information.

Port notation uses the following convention: number of ports (number of clock lanes x number of data lanes). For example, the port notation for a serializer with two ports that each contain four data lanes is expressed as 2(1x4). This can be further condensed to 2x4. The number of data lanes in this notation is the maximum available; the ports can be optionally configured to use less than the maximum. Four-lane ports can operate using one, two, three, or four data lanes.

ADI Confidential for New vision under NDA
Analog Confidential Distribute only under NDA
Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA

The DSI serializers feature MIPI D-PHY version 1.2 and MIPI DSI version 1.3. A simplified block diagram (Figure 142) shows the video path from the dual port DSI serializer input to the dual serial link output. The functional block diagrams (Figure 143 and Figure 144) show the MIPI ports, DSI controllers, and video pipe routing in more detail.

Note: The DSI ports output PPI signaling to the DSI Rx Controller. Reference the MIPI D-PHY specifications for additional information. DPI is an internal parallel video bus used to send the video from the DSI controllers to the video pipes. The PCLK is derived from the MIPI Clock; the PCLK generation block feeds into both the DSI Rx Controller and the GMSL link.

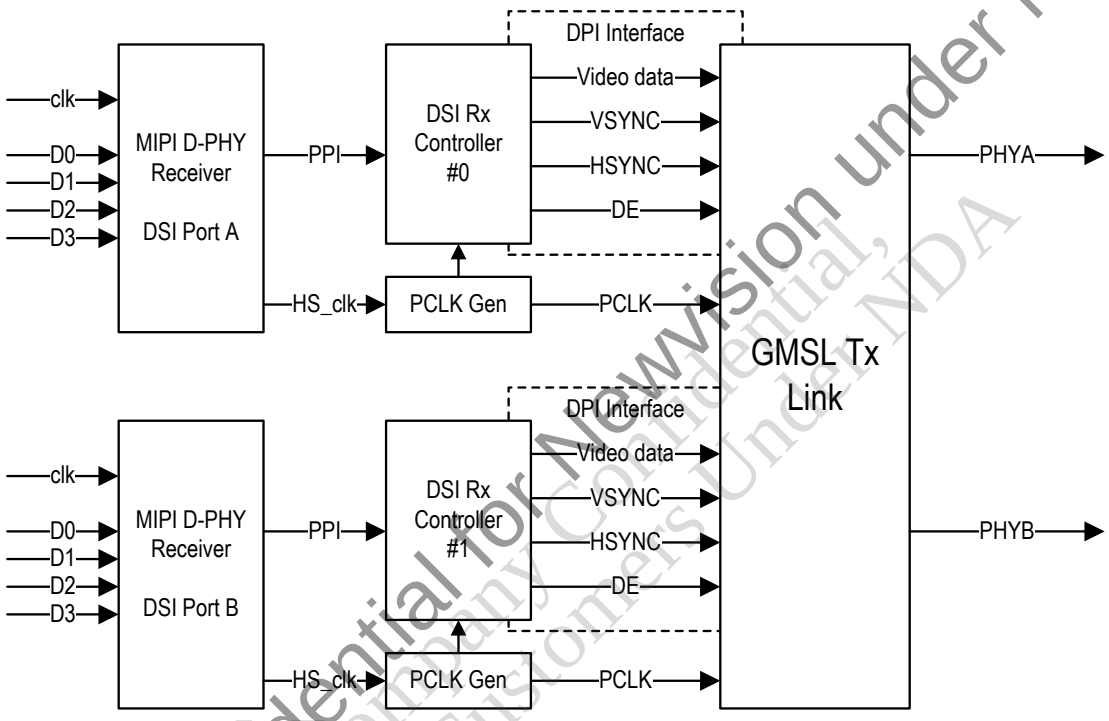


Figure 142. Simplified Block Diagram of Dual Port DSI Serializer

ADI Confidential for Newvision under NDA
Analog Devices, Inc.
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

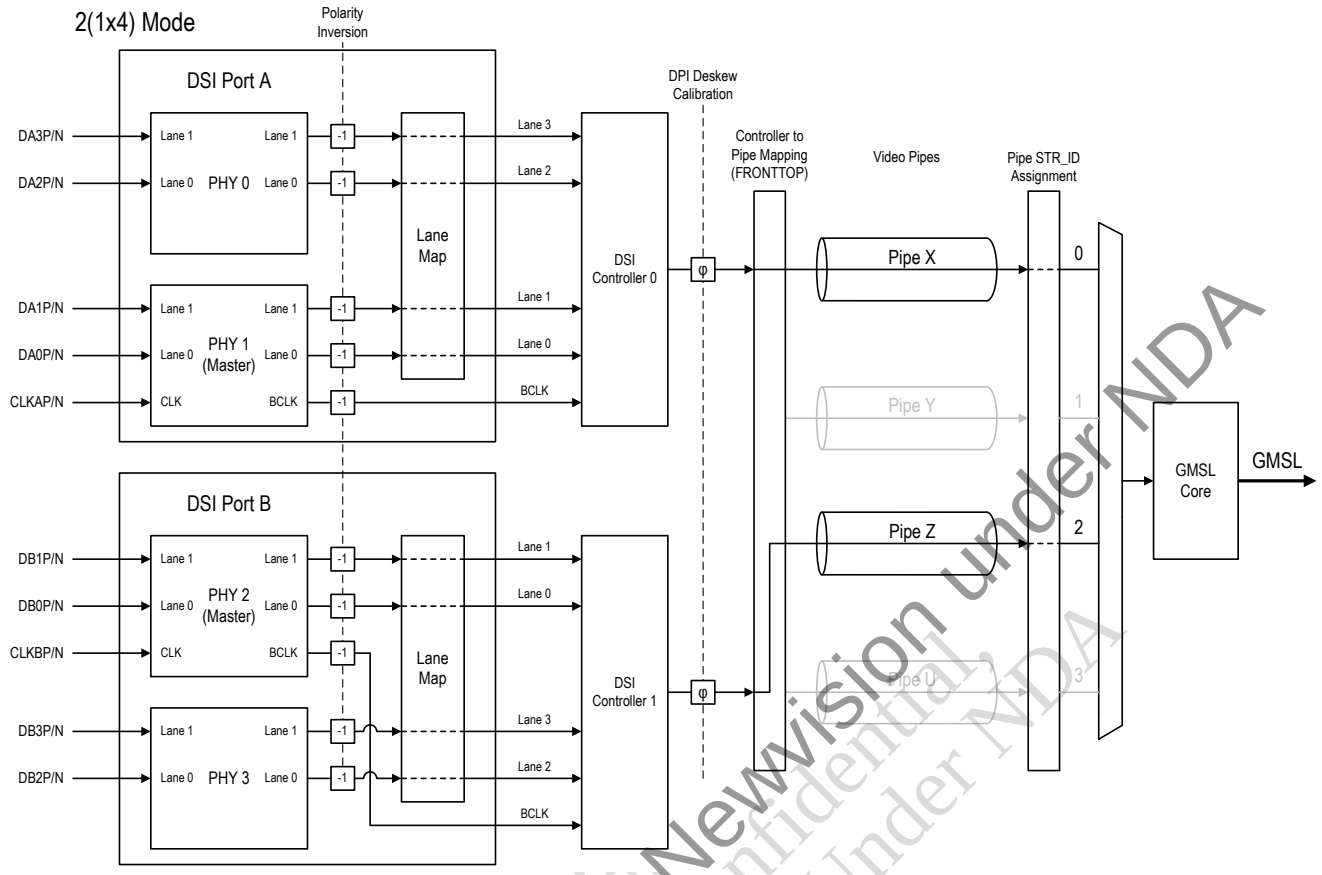


Figure 143. GMSL2 DSI Serializer – 2(1x4) Mode

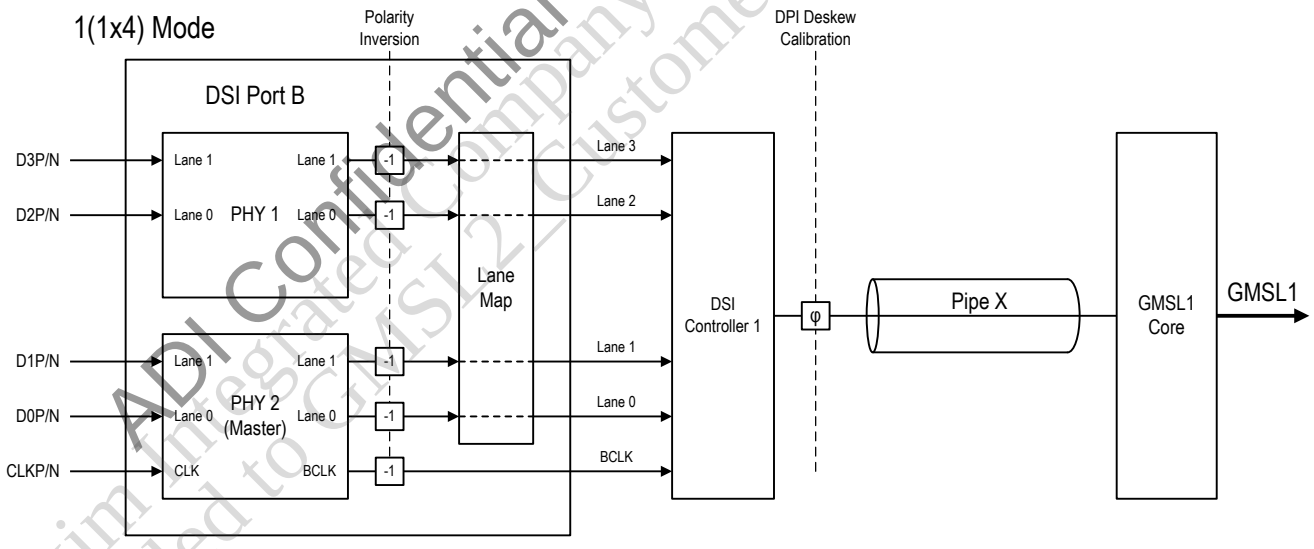


Figure 144. GMSL2 DSI Serializer – 1(1x4) Mode (MAX9283/85)

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.2.1 DSI Rx Controller

The DSI Rx controller consists of four stages: Lane Merger, Packet Processor, DPI Constructor, and DPI Deskew. A simplified block diagram of the DSI Rx controller is presented in Figure 145.

- **Lane Merger**: Receives the incoming data from multiple D-PHY data lanes and merges the data to a single packet.
- **Packet Processor**: Receives the merged packet and extracts the necessary information (e.g., data type, packet length, and payload). The packet processor also checks the integrity of the packet header ECC and payload CRC; any ECC or CRC errors are reported to ERRB.
- **DPI Constructor**: Receives the processed packet data and generates parallel Vertical Sync (VS), Horizontal Sync (HS), Data Enable (DE), and 24-bit video data.
- **DPI Deskew**: Removes any jitter from the processed HS and VS signals from the DPI constructor.

Note: Virtual channel routing is not supported by GMSL2 DSI serializers.

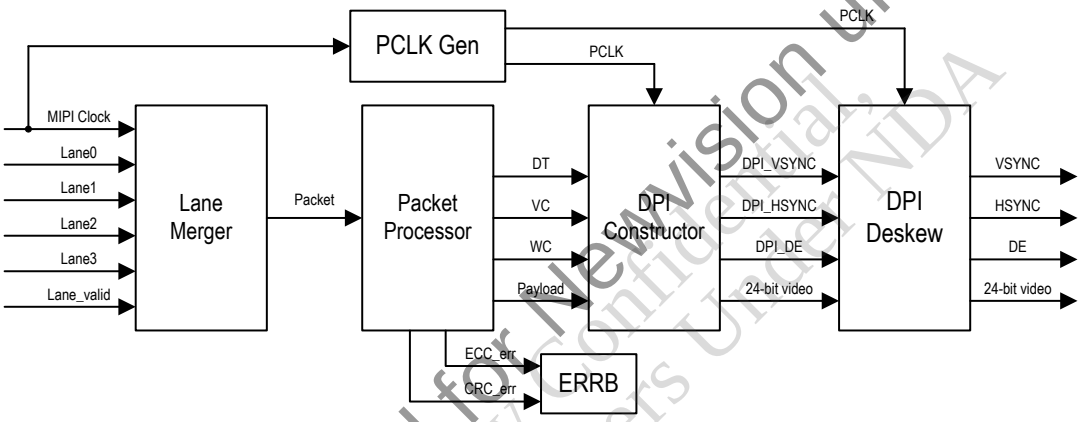


Figure 145. Simplified Block Diagram of DSI Rx Controller

40.2.1.1 Packet Processor

The packet processor handles the merged MIPI lane data according to the DSI protocol specification.

40.2.1.1.1 MIPI DSI Protocol

The DSI protocol converts video data into packets and transmits the packets through MIPI D-PHY interface. There are two modes of data transmission: High Speed (HS) and Low Power (LP). GMSL2 DSI serializers support only HS data transmission mode.

The DSI serializers allow multiple packets to be concatenated into one HS transmission. This concatenation results in an effective increase in bandwidth by eliminating time gaps between packets and reducing the number of transmissions required.

MIPI D-PHY v1.2 and MIPI DSI v1.3 define a dedicated End of Transmission Packet (EoTp) that signals the end of an HS transmission. The EoTp detection can be disabled in the serializer for backward compatibility with earlier MIPI specifications.

Figure 146 below shows an example of an HS burst transmission with EoTp enabled.

Analog Devices Confidential for New Revision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

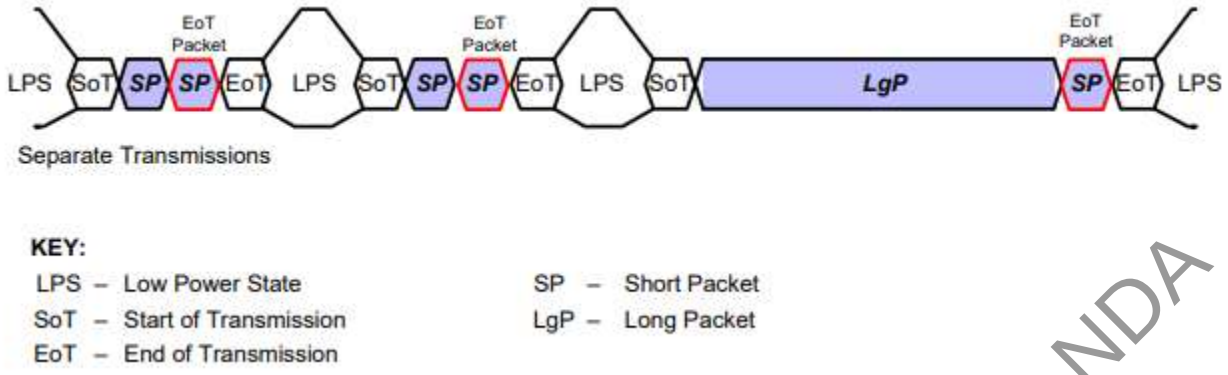


Figure 146. Example of HS Burst Packets with EoTp

40.2.1.1.2 MIPI DSI Packets

There are two types of DSI packets: long packets and short packets. Long packets (Figure 147) are primarily used for large blocks of video or other data. Short packets (Figure 148) are used to convey accurate timing information (e.g., HS or VS edges).

Long packets consist of a packet header (PH), payload, and packet footer (PF):

- The packet header contains a 6-bit data type, a 2-bit virtual channel number, a 16-bit word count, and an 8-bit error correction code (ECC).
- The payload length is determined by the 16-bit word count field in the packet header.
- The packet footer contains a 16-bit checksum (CRC).

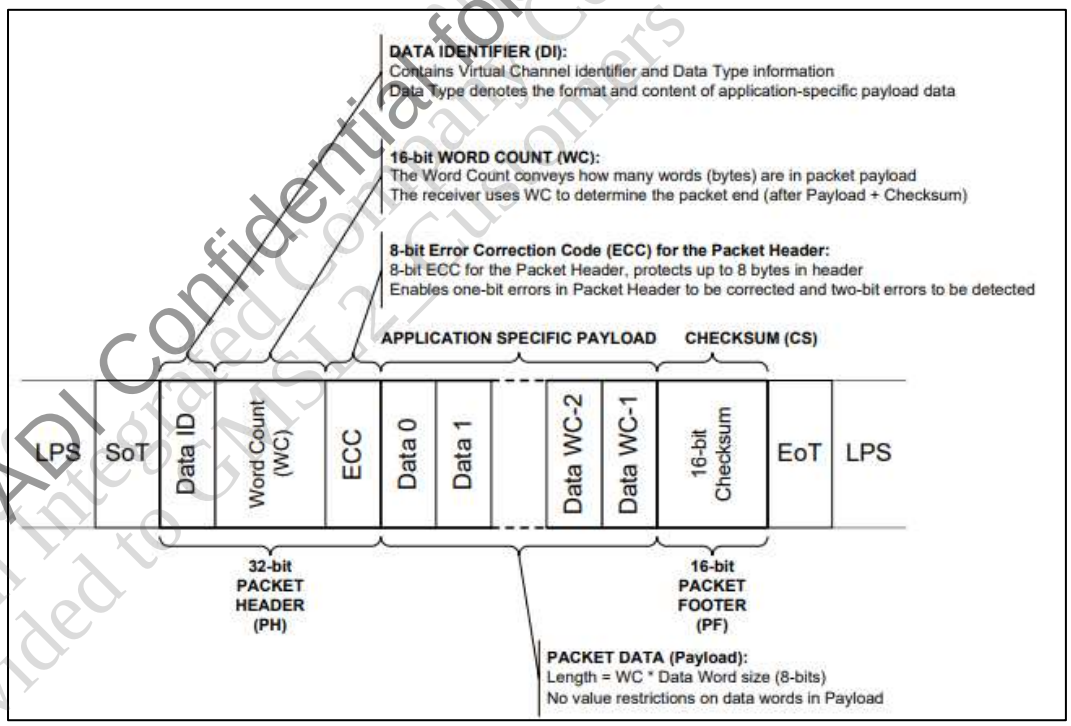


Figure 147. DSI Long Packet Structure

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribution only to Maxim Integrated Company Customers Under NDA
Provided to GMSL2_Customers Under NDA

Short packets consist of a 6-bit data type, a 2-bit virtual channel number, a 16-bit packet data, and an 8-bit ECC.

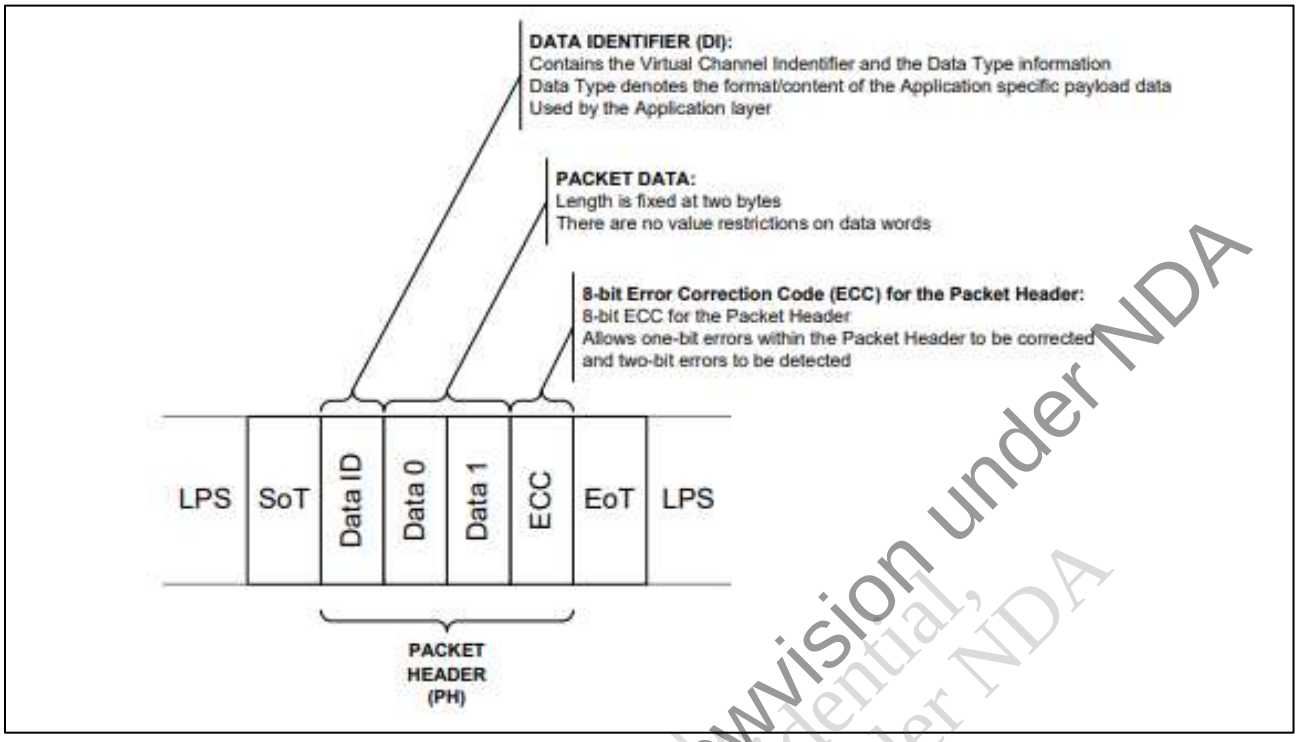


Figure 148. DSI Short Packet Structure

GMSL2 DSI serializers support the long and short packets shown below in *Table 174*.

Table 174. Supported MIPI DSI Packets

Data Type (HEX)	Data Type (BIN)	Description	Packet Size
0x01	00 0001	Sync Event, V Sync Start (VSA)	Short
0x11	01 0001	Sync Event, V Sync End (VSE)	Short
0x21	10 0001	Sync Event, H Sync Start (HSA)	Short
0x31	11 0001	Sync Event, H Sync End (HSE)	Short
0x08	00 1000	End of Transmission packet (EoTp)	Short
0x3E	11 1110	Packed Pixel Stream, 24-bit RGB888	Long
0x09	00 1001	Null packet, no data	Long
0x19	01 1001	Blanking packet, no data	Long

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Confidential, Provided to GMSL Customers Under NDA

40.2.1.1.2.1 Sync Events (Short Packets)

Sync event are short packets that include Vertical Sync Start, Vertical Sync End, Horizontal Sync Start, and Horizontal Sync End. These packets are used to define video timings, including the lengths of the vertical/horizontal sync pulses, front porch, or back porch.

Each image frame begins with a VS Start packet; each line begins with a HS Start packet.

The VS End and HS End packets are used only in the Sync Pulse video mode (refer to the DPI Constructor section below for more information). For Sync Event video mode, only VS Start and HS Start packets are required.

40.2.1.1.2.2 RGB888 (Long Packet)

RGB888 video data is transmitted using the 24-bit Packed Pixel Stream data type (0x3E). The pixel format is ordered Red (8 bits), Green (8 bits), and Blue (8 bits), as illustrated below in Figure 149.

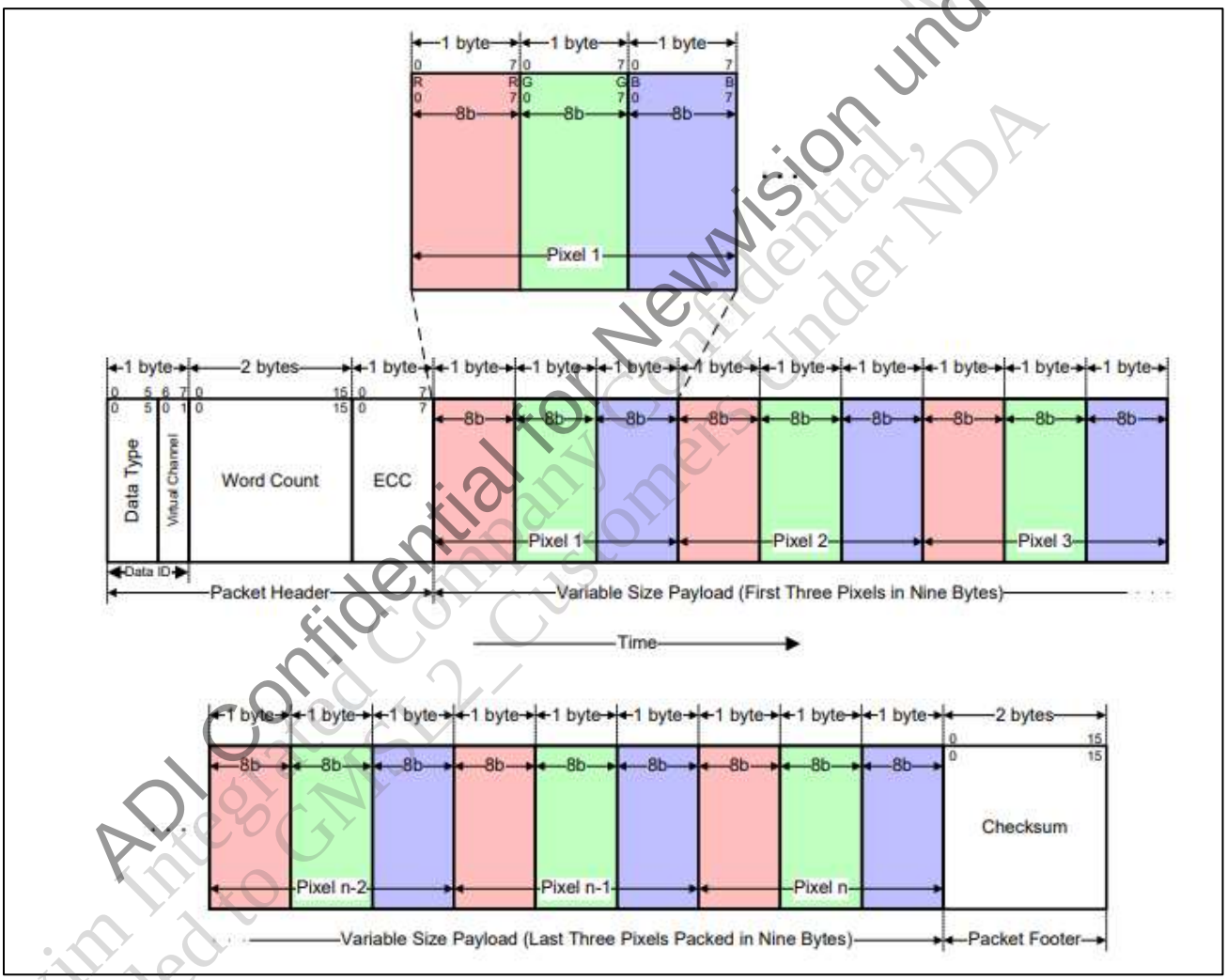


Figure 149. Color Format of RGB888

ADI Confidential for NewVision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.2.1.2 DPI Constructor

The DPI Constructor builds the DPI video signals (HS/VS/DE/24-bit video) based off video timings derived from the Sync Event short packets, RGB888 long packets, and the Blanking/Null long packets.

40.2.1.2.1 Video Timing Modes

GMSL2 DSI serializers support two video timings modes:

- Non-Burst Mode with Sync Pulses (*Figure 150*).
- Non-Burst Mode with Sync Events (Figure 151).

Explanation of abbreviations used in *Figure 150* and *Figure 151*:

- VSS: VS Start Packet
- VSE: VS End Packet
- HSS: HS Start Packet
- HSE: HS End Packet
- RGB: RGB888 Long Packet
- VACT: Active Video Data
- VSA: Vertical Sync Active
- VBP: Vertical Back Porch
- VFP: Vertical Front Porch
- HSA (Blanking Packet): Horizontal Sync Active or Low Power Mode, no Data
- HBP (Blanking Packet): Horizontal Back Porch or Low Power Mode, no Data
- HFP (Blanking Packet): Horizontal Front Porch or Low Power Mode, no Data
- BLLP (Blanking Packet): Horizontal Sync non-Active, Vertical Back Porch, Vertical Front Porch or Low Power Mode, no Data

ADI Confidential for NDA vision under
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.2.1.2.1.1 Non-Burst Mode with Sync Pulses

Defines video timing by sending both Sync Start and Sync End packets (Figure 150). The DSI serializer receives the short packets (HS/VS) and long packets (RGB888) to generate HS/VS/DE signals.

- Vertical Sync (VS) signal is set HIGH when VS Start packet is received.
- VS is set LOW when the VS End packet is received.
- Horizontal Sync (HS) signal is set HIGH when the HS Start packet is received.
- HS is set LOW when the HS End packet is received.
- The Data Enable (DE) signal is set HIGH when the RGB888 datatype Long Packet is received with valid video data.

This mode allows the serializer to accurately reconstruct the original video timing and maintains the sync pulse widths transmitted by the source.

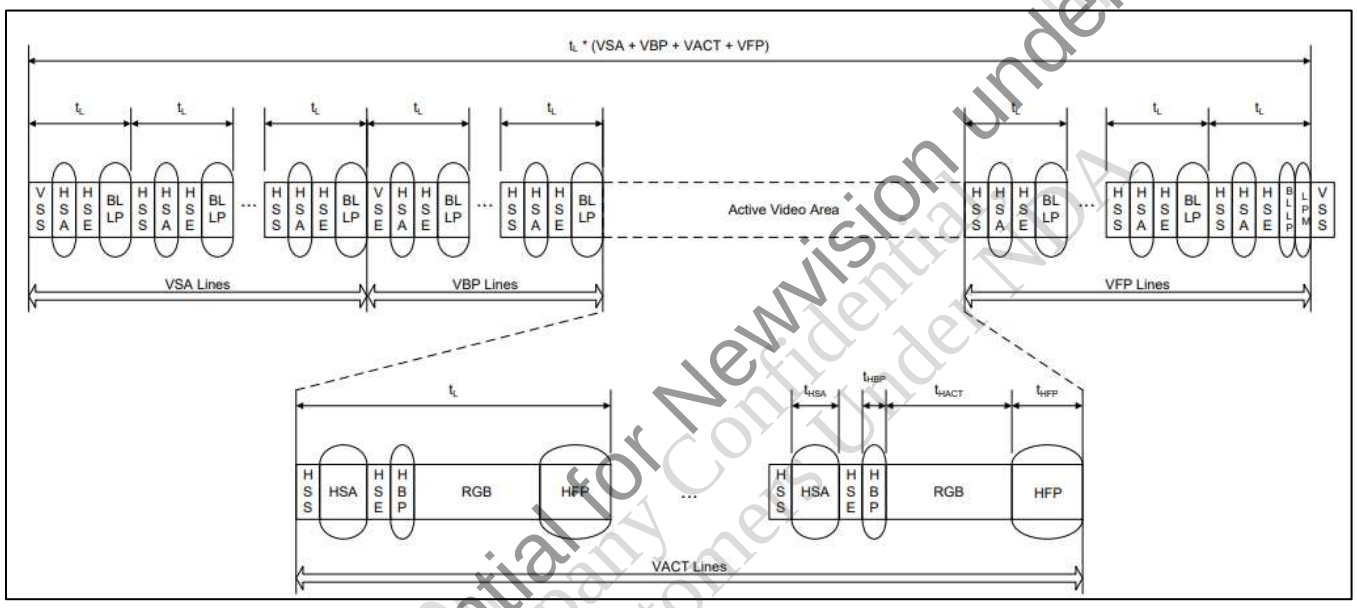


Figure 150. Non-Burst Mode Transmission with Sync Start and Sync End (Sync Pulses)

ADI Confidential for Newvision under NDA
Analog Devices Confidential under NDA
Maxim Integrated Company Confidential under NDA
Provided to GMSL2_Customers

40.2.1.2.1.2 Non-Burst Mode with Sync Events

Defines video timing using only Sync Start packets (Figure 151). Accurate sync pulse width is not required with this mode.

- Vertical Sync (VS) signal is set HIGH when VS Start packet is received.
- VS pulse width is defined by the register cfg_dpi_VS_width_h and cfg_dpi_VS_width_l.
- Horizontal Sync (HS) signal is set HIGH when the HS Start packet is received.
- HS pulse width is defined by the register cfg_dpi_HS_width_h and cfg_dpi_HS_width_l.
- The Data Enable (DE) signal is set HIGH when the RGB888 datatype Long Packet is received with valid video data.

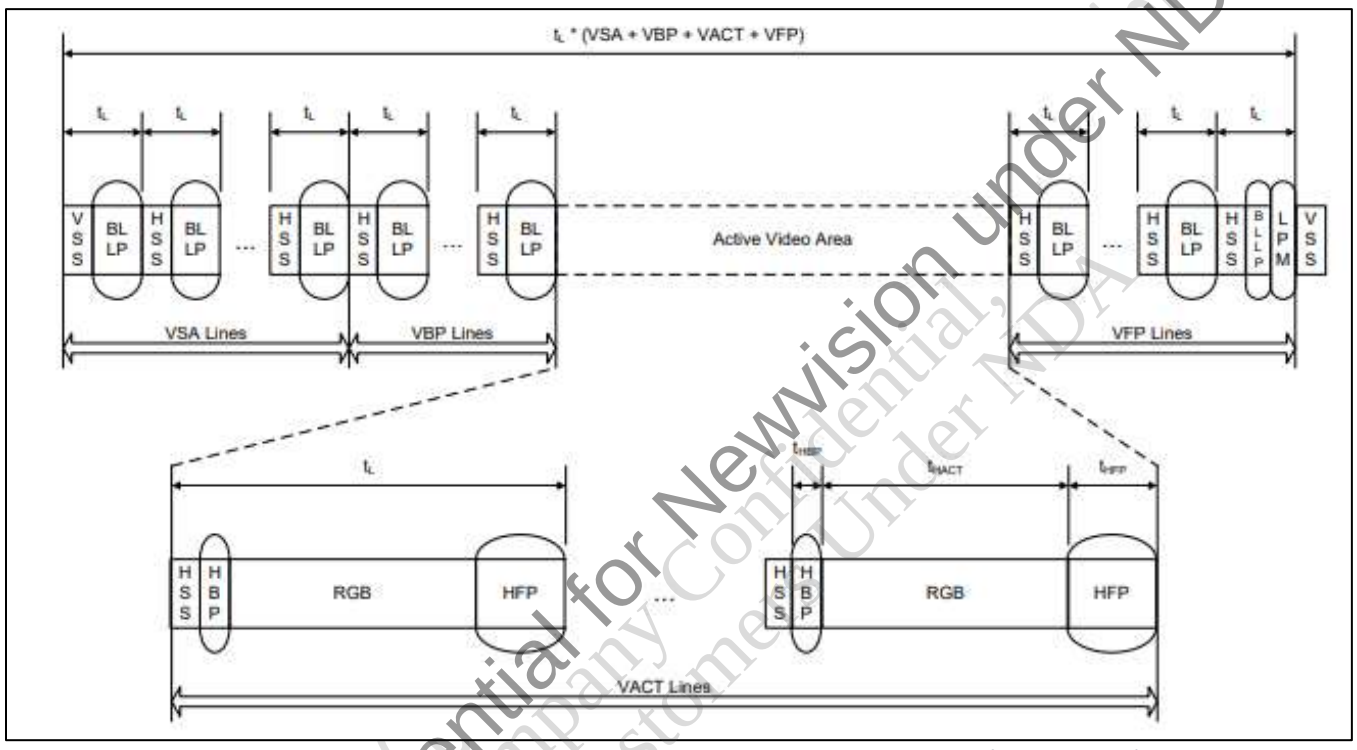


Figure 151. Non-Burst Mode Transmission with Sync Start Events (Sync Events)

Data Enable (DE) is generated when an RGB888 long packet is received with valid video data. When DE is high, 24-bit RGB888 video data is sent out through the DPI port. The DE signal is ACTIVE HIGH.

VS and HS polarity can be inverted by programming the internal registers cfg_dpi_VS_polarity and cfg_dpi_HS_polarity.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Under NDA
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2 Customers Under NDA

40.2.1.2.1.3 Continuous Clock Mode

GMSL2 DSI serializers only support continuous clock mode of operation. To ensure compatibility, a typical MIPI D-PHY transmitter must follow the below sequence:

- Initiate the MIPI D-PHY transmitter in a STOP state (LP11).
- Wait for the DSI serializer to be powered up and configured.
- Start the clock lane by initiating a start-of-transmission (SOT) sequence followed by maintaining the continuous clock mode in high-speed (HS) mode.
- The clock will remain in HS mode once it enters the HS mode; the clock will stay in HS mode until the chip is powered down.

The DSI controller operates on two different clocks: the MIPI Byte clock and the pixel clock (PCLK). The MIPI Byte clock is generated by the MIPI clock lane receiver. The PCLK is derived from the incoming MIPI clock (not MIPI Byte clock) with a frequency fast enough to clock the video data at the same data rate as the MIPI DSI input. If the MIPI clock stops, both the MIPI Byte clock and the PCLK will also stop. This will cause the DSI controller to cease operation.

40.2.1.3 DPI Deskew

The VS and HS signals output from the DSI controller may incur ±2 PCLK jitter resulting from the clock domain transitioning from the received MIPI clock to the parallel pixel clock domain (i.e., independent of jitter from the incoming MIPI video stream). This jitter will be further influenced by timing inconsistencies in the MIPI data stream. Jitter can cause image disruption for display modules and/or controllers that require precise blanking time(s). Note that this is not an issue for display modules using only DE.

The DPI Deskew feature is designed to remove the frame-to-frame timing jitters that may be present in the sync signals. The DPI Deskew block removes jitter on the sync signals by reconstructing the sync signals based on timings provided (i.e., programmed to registers) by the user. After detecting the first incoming frame, the deskew block begins to generate sync signals and output video data in accordance with the users' timings. Correct configuration and timing information is necessary for the deskew feature to work effectively.

The DSI Deskew block requires that the average frame period of the incoming video to be fixed and that the frame-to-frame jitter to be less than the Deskew FIFO size (in units of PCLKs). If the average frame-to-frame jitter is larger than the FIFO size, or the frame-to-frame jitter accumulates (instead of averaging to zero), the jitter can overflow the FIFO block and cause the block to stop and wait for the next frame.

The SoC (DSI Source) should synchronize the DSI Tx Controller and MIPI D-PHY Transmitter to meet the jitter requirement. All lines of the frame (i.e., active video lines and vertical blanking lines) should be the same fixed length.

Note: If the incoming MIPI data stream is excessively inconsistent, the DPI Deskew feature may not be compatible.

ADI Confidential for review only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customer under NDA

40.2.1.4 Error Reporting

MIPI DSI errors are reported in the dsi_err_h and dsi_err_l registers. When an error is detected, the bit corresponding to the error will be latched until it is read out via a control channel. Typical errors reported are ECC and CRC errors.

40.2.1.4.1 ECC Errors

The DSI controller checks for 1-bit and 2-bit ECC errors within of the packet header. Refer to Figure 147 above for the structure of a long packet.

- If a long packet has a 1-bit error in the data ID or word count of the header, the DSI controller corrects the header and will also report a 1-bit ECC error (dsi_err_l[3] = 1).
- If a long packet has a 2-bit error in the data ID or word count of the header, the DSI controller will report a 2-bit error (dsi_err_l[5] = 1), but will not recover the data ID or word count.
- If there is a 1-bit error in the ECC area of the packet header, the DSI controller will not correct the header, but will report the 1-bit ECC error (dsi_err_l[4] = 1).
- If there are 2-bit errors in the ECC area of the packet header, both register bits may be set HIGH (dsi_err_l[5:4] = 1).

40.2.1.4.2 CRC Errors

The DSI controller also checks the CRC of the packet payload and will report if a CRC error is detected. A CRC error is detected when either of the following occur:

- there are error bits in the payload, or
- errors in the packet header cause the DSI controller to derive an incorrect word count from the header.

A CRC error is signaled when dsi_err_l[6] = 1.

ADI Confidential for Newvision Under NDA
Analog Devices Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.3 Configuration

A programming flowchart for GMSL2 DSI serializers is presented in Figure 152. Each step is explained in further detail in corresponding sections.

Note: The programming flowchart in *Figure 152* is comprehensive. Actual programming process will vary by device-specific feature availability. Reference *Feature Availability by Device* and device datasheets for more information.

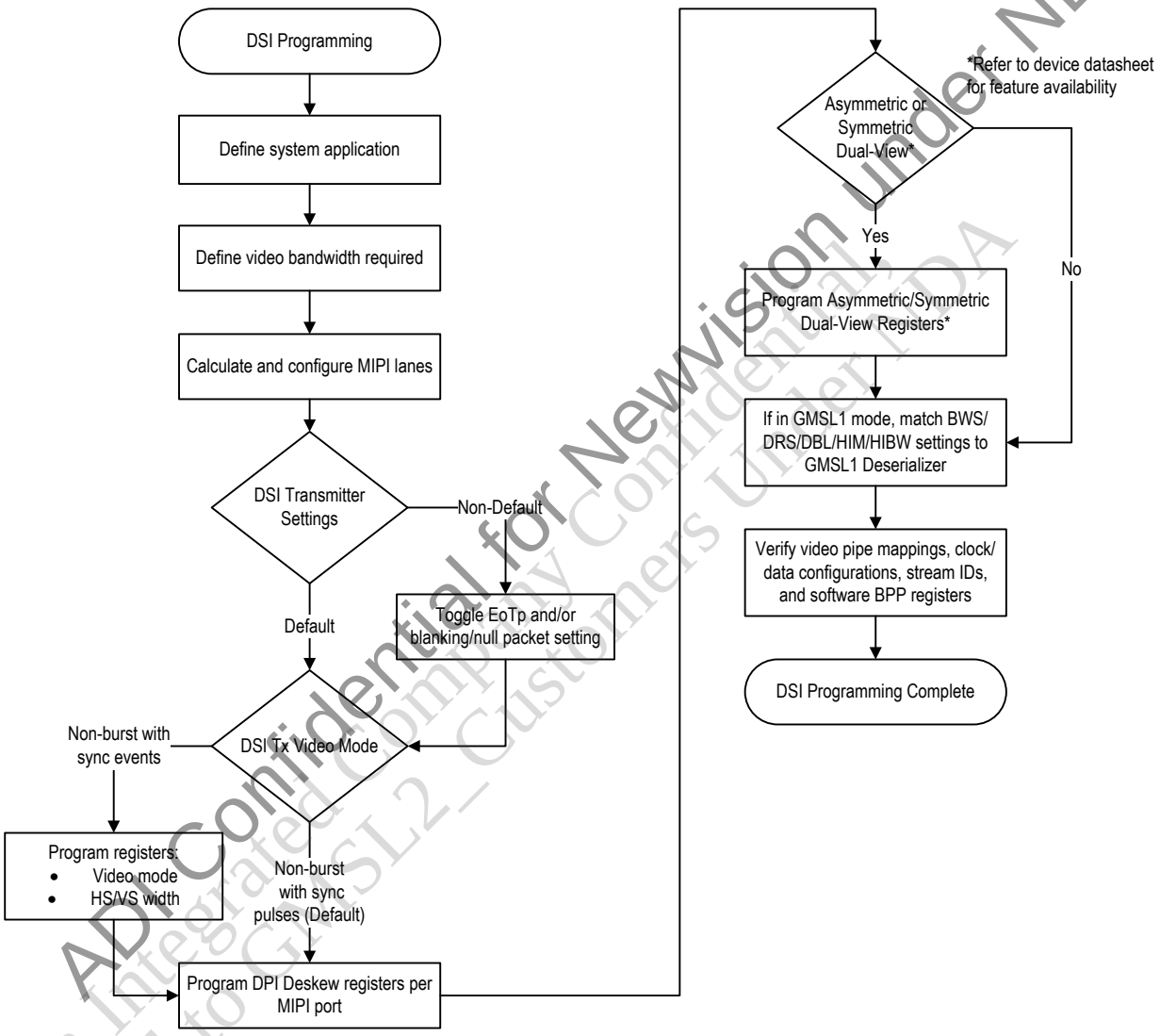


Figure 152. DSI Configuration Flowchart

ADI Confidential for NewVision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

40.3.1 Modes of Operation

The dual-view applications supported by each DSI serializer part family are shown below in Table 175.

Table 175. DSI GMSL Supported Applications

GMSL2 DSI Serializers	GMSL PHY Combinations	Applications Supported
MAX9283 **MAX9285**	1x GMSL1	Single Display (GMSL1)
MAX96755/F **MAX96757/F**	1x GMSL1 2x GMSL2	Single Display (GMSL1) Dual Independent Displays, Duplicated Displays, Symmetric Dual-View (GMSL2)
MAX96789/F **MAX96791/F**	2x GMSL1 2x GMSL2 1x GMSL1 and 1x GMSL2	All available applications, single/dual/symmetric/asymmetric in any combination of GMSL1/GMSL2 PHY's

Depending on application and use-case, GMSL2 DSI serializers may require significant register programming. Basic applications are covered below. Identify the relevant use-case and proceed to the appropriate section(s) below for device programming support.

Operation Considerations:

- The device must be configured before receiving any data. Dynamic configuration is not supported.
- In accordance with the DSI standard, the Tx clock and data lanes require special start and stop pattern to set DSI receiver into HS/LP mode.
- Whenever the DSI serializer is reset or power cycled, first reconfigure the GMSL system, then restart the DSI source.
- The abnormal glitch on the MIPI TX data lanes created by DSI source at power up can cause abnormal operation and/or MIPI errors on serializer.
  It is highly recommended to apply below MIPI receiver reset routine in MIPI serializers after every DSI source power up.
  - Power up DSI source, wait for MIPI output to be stable at LP11 (wait time depends on MIPI source and should be set based on the system)
  - Set mipi_rx_reset (register 0x330 in all DSI Serializers) to '1'
  - Set mipi_rx_reset (register 0x330 in all DSI Serializers) to '0'
  - Start DSI video streaming
- The DSI serializer accepts only 24-bit data.
- The MIPI receiver only operates in the continuous clock mode.
- Burst mode and command mode operation are not supported.
- Video data must pass through pipe X to reach the dual-view / asymmetric engine block.
- Video data must pass through pipe X (or pipe Y, if available on the PHY) for HDCP to function.
- GMSL1 operation requires that GMSL PHY A receives video from pipe X, and GMSL PHY B (if available) receives video from pipe Y.
- Four-lane ports can operate using one, two, three, or four data lanes.

ADI Confidential for Newvision Under NDA
Analog Devices Confidential, Distribution Only to Maxim Integrated Employees and Customers Under NDA
Provided to CMSL2 Customers Under NDA

40.3.1.1 Single Display

A single MIPI DSI source sends a single video to a single MIPI DSI port of a GMSL2 DSI serializer (Figure 153). The video is sent across a single GMSL link.

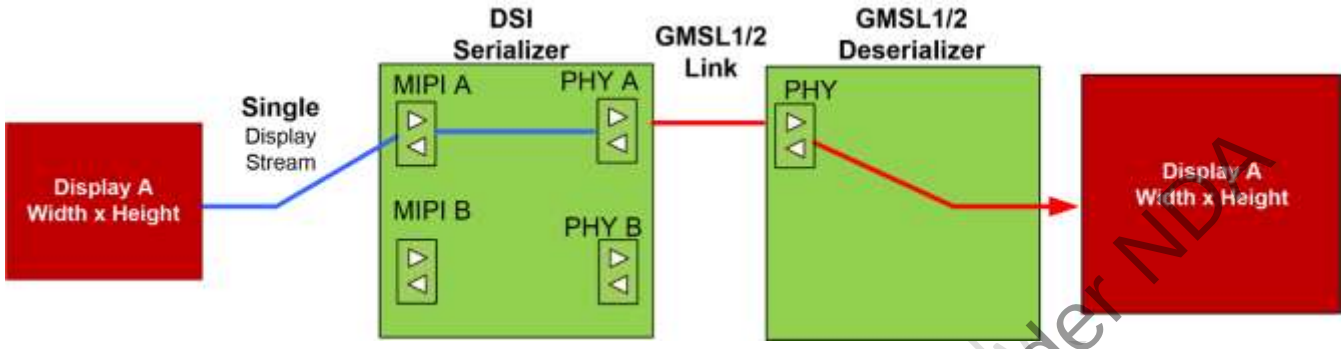


Figure 153. Single Display Application

40.3.1.2 Dual Independent Displays

Two MIPI DSI sources send two individual videos to two independent MIPI DSI ports of a GMSL2 DSI serializer (Figure 154). Note that this application is limited by the available MIPI ports and GMSL PHYs. The videos do not have any restrictions nor relationship to each other (i.e., the framerate, resolution, and timings are independent).

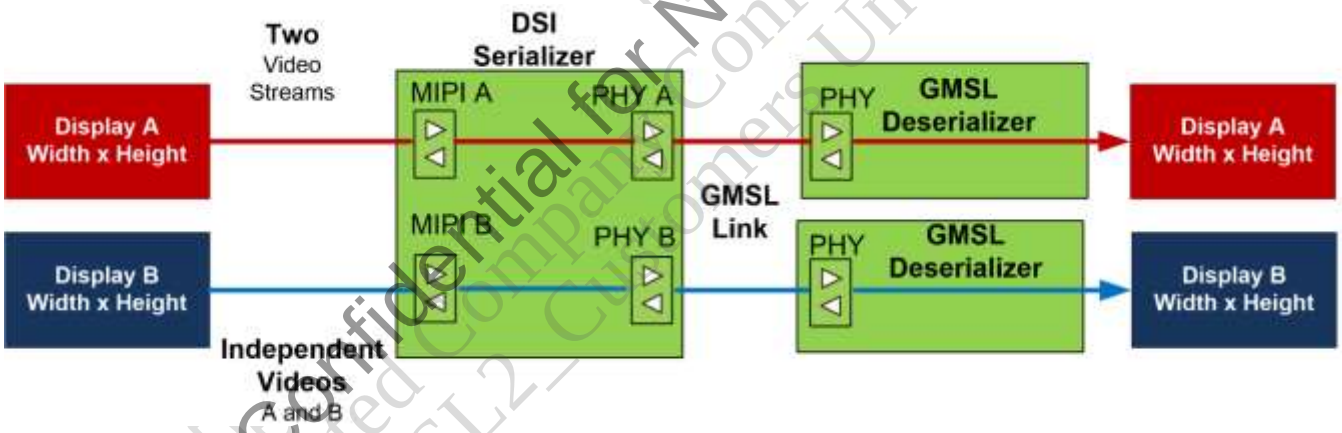


Figure 154. Dual DSI Independent Displays Application

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.3.1.3 Duplicated Display

A single MIPI DSI source sends a video to a single MIPI DSI port; the video is then sent to two deserializers from different GMSL PHYs (Figure 155).

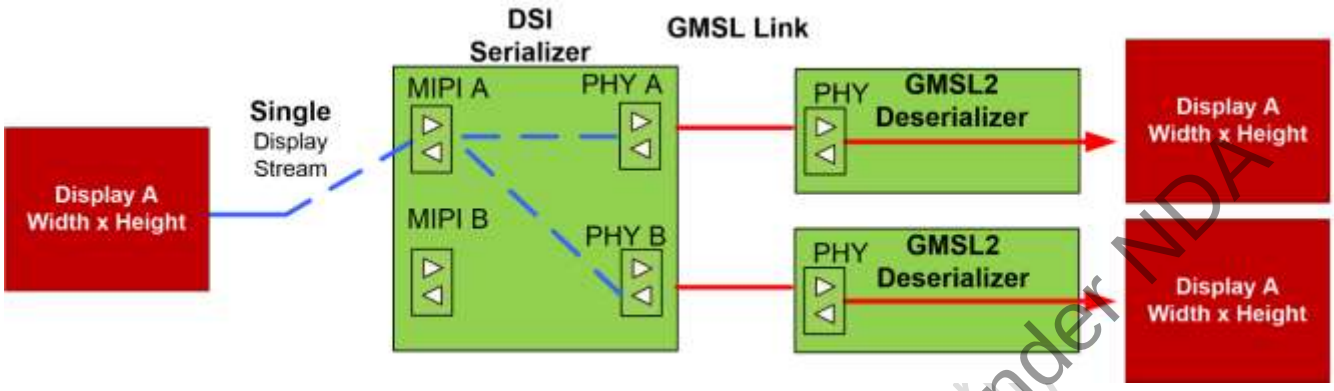


Figure 155. Duplicated Display Application

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute Only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.3.1.4 Symmetric Dual-View

The SoC sends a “superframe” using a single MIPI DSI port. The DSI serializer receives the single video, splits the video horizontally into two videos, and transmits each half of the video over an independent GMSL link (Figure 156).

Note: A “superframe” consists of two symmetric side-by-side, column interleaved, or line alternating videos. Reference the *Dual-View* section for additional information.

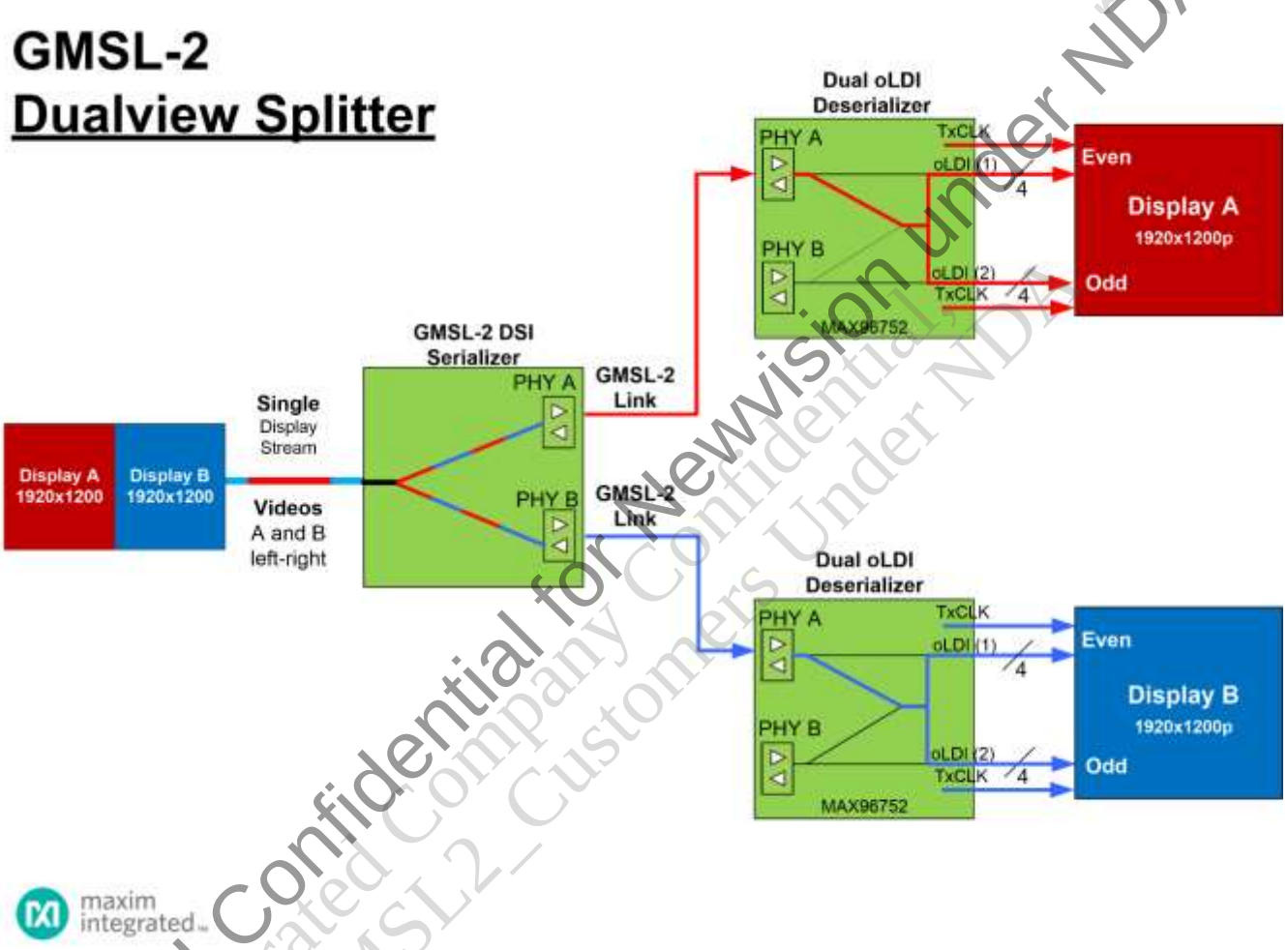


Figure 156. Symmetric Dual-View Splitter Application

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

40.3.1.5 Asymmetric Dual-View

The SoC sends an “asymmetric superframe” consisting of two images of different resolutions embedded into a single image to a single MIPI DSI port. The DSI serializer receives the single video, splits the video into two videos, and transmits the separate videos over independent GMSL links.

Asymmetric dual-view is depicted in Figure 157. In the example, the smaller image on the right has dummy lines that will be removed by the serializer to effectively recreate a smaller image resolution after passing through the asymmetric engine of the asymmetric DSI serializer.

The requirements for asymmetric dual-view are much more restrictive than symmetric dual-view. Care should be taken this application, and any potential incompatibilities must be considered. Not all DSI video sources can incorporate dummy lines into a video nor manage two videos in a single stream.

Note: A “superframe consists” of two symmetric side-by-side, column interleaved, or line alternating videos. Reference the *Dual-View* section for additional information.

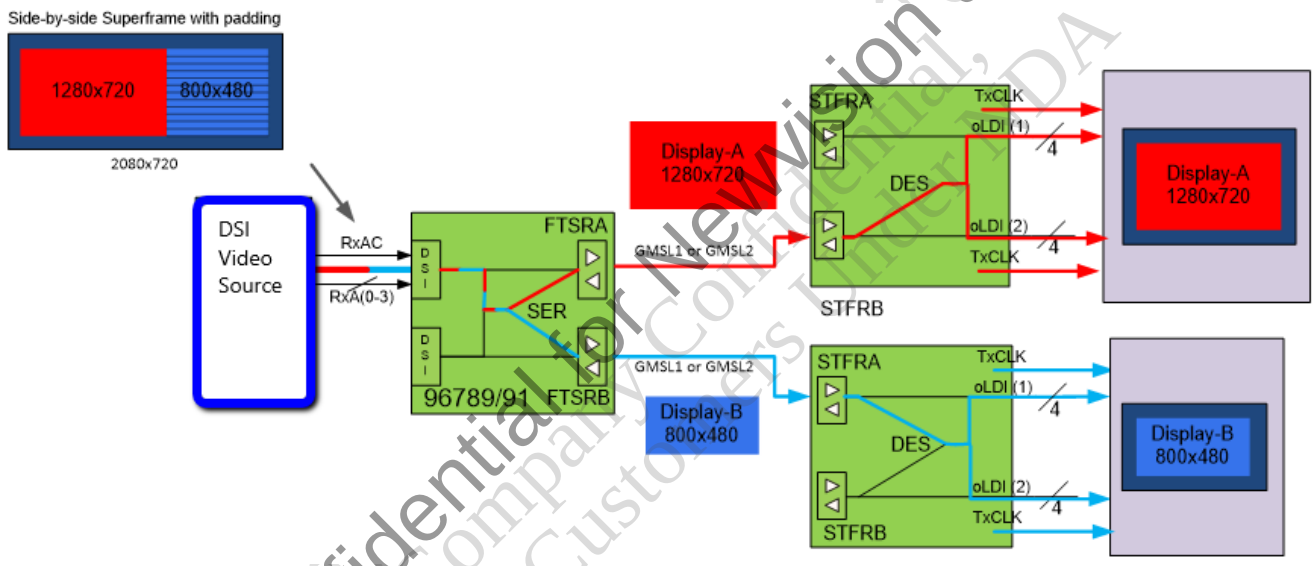


Figure 157. Asymmetric Dual-View Splitter Application

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.3.1.6 Dual Port "Ganged Mode"

The SoC sends a single video split across two MIPI DSI ports (Figure 158). The video is divided by odd/even pixels, with odd pixels sent to Port A and even pixels sent to Port B. The odd/even pixels are then brought back together in the Ganged Mode block, and a common PLL is used to recover the required pixel clock.

Ganged mode is used when the bandwidth requirement of the single video source exceeds the capability of the DSI port.

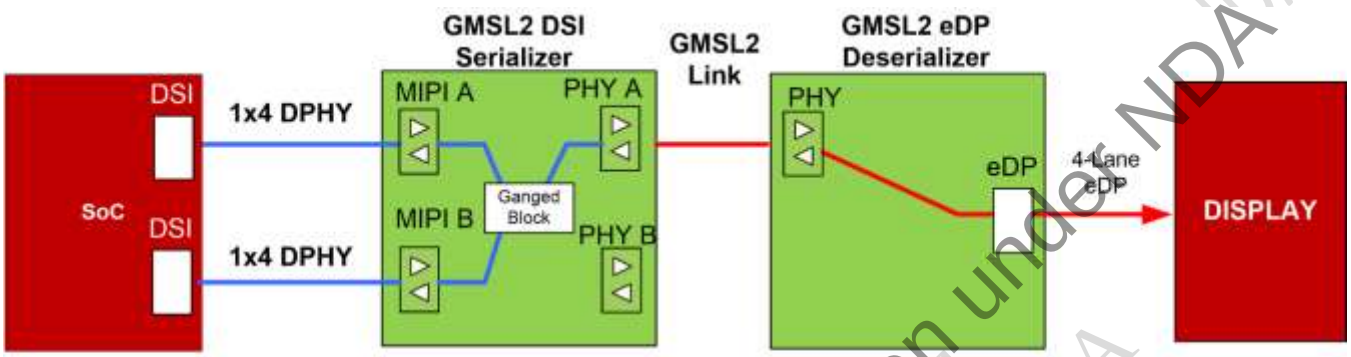


Figure 158. Dual Port "Ganged Mode"

ADI Confidential for Newvision under NDA
Analog Devices Confidential,
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.3.2 MIPI Video Bandwidth

The video bandwidth is calculated using the video resolution, blanking times, and frame rate. When configuring the DSI video source, use the slowest lane rate possible which supports the video bandwidth (with sufficient margin). This results in:

- reduced MIPI bandwidth results in lower MIPI frequency and improved signal integrity, and
- increased accuracy of the timing relationship of input PCLK to output MIPI clock.

The number of MIPI lanes used contributes to the overall MIPI video payload. The following equations can be used to calculate the MIPI lane rate(s):

$$PCLK = (Total\ Horizontal) \times (Total\ Vertical) \times (Frames\ Per\ Second)$$

$Video\ Payload = PCLK \times 24\ (bpp)$, must not exceed 5.2Gbps due to link overhead

$MIPI\ Lane\ Rate = \frac{PCLK \times bpp}{\#\ of\ Lanes}$, four possible lanes (see *Table 176* for maximum lane rates)

The MIPI video payload can be derived by the video throughput and number of lanes used. For a 6Gbps serial link, the maximum video throughput is 5.2Gbps (due to 9b10b encoding overhead and guard band protection). If using four lanes, 5.4Gbps/4 = 1.35Gbps per lane; this limit should be further rounded down to 1.3Gbps per lane to provide margin for other data channels. Each MIPI lane supports up to 2.5Gbps. See *Table 176* for maximum MIPI lane rates which fit within a 6Gbps GMSL link.

Reference the GMSL2 Link System Bandwidth section for information regarding other data channels and overall serial link bandwidth.

Table 176. Maximum MIPI Lane Rates for 6Gbps Link

Number of MIPI Lanes	Maximum MIPI Lane Rate(s) for a 6Gbps link
4	1.3Gbps
3	1.7Gbps
2	2.5Gbps
1	2.5Gbps

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

40.3.2.1 DSI vs. CSI-2

- DSI is Display Serial Interface and is focused on display applications; CSI-2 is Camera Serial Interface and is focused on camera applications.
- GMSL2 DSI serializers accept only 24-bit RGB888 color.
- The DSI and CSI-2 packet-level protocols are different and not interchangeable. A DSI transmitter must be matched to a DSI receiver; a CSI-2 transmitter must be matched to a CSI-2 receiver.
- The physical layer (PHY) is the same for most cases, but packet processing will not work for mismatched MIPI interfaces.
- All MIPI devices operate in the continuous clock mode only.

The MIPI Rx DSI interface receives MIPI data and converts it to parallel data to be sent into the video pipelines. The data is then serialized and sent across the link. The video pipes only contain generalized video pixel data and do not retain MIPI data.

40.4 Reference the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

40.4.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable

Maxim Confidential, Provided to GMSL2_Customers_Under_NDA
Advance Information Preliminary Product Specification
Not to be copied without permission
Advance Information Preliminary Product Specification
Not to be copied without permission
Maxim Confidential, Provided to GMSL2_Customers_Under_NDA

selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)

Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Vision under NDA

0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock, lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

40.4.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for New Version Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

40.5 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 section for more information.

40.5.1 MIPI Input Configuration

Most DSI serializers have two MIPI ports. Each MIPI port has an associated MIPI controller with dedicated registers.

- MIPI Port A — DSI controller 0
- MIPI Port B — DSI controller 1

The number of MIPI lanes per port is configurable by changing registers below. The default is four lanes for each port.

- Port A:
  - 0x331[1:0]: ctrl0_num_lanes[1:0]
- Port B:
  - 0x331[5:4]: ctrl1_num_lanes[1:0]

Use the equation below to program the DSI transmitter lane rate in the video source. DSI serializers only accept 24-bit RGB888 datatype; the bpp value is always equal to 24.

$$MIPI\ Lane\ Rate = \frac{Pixel\ Clock \times Bits\ Per\ Pixel}{\#\ of\ Lanes}$$

40.5.1.1 Lane Mapping

All DSI serializers are capable of 2(1x4) MIPI configuration (i.e., two independent MIPI ports, each with four differential data lanes and one differential clock lane) except for MAX9283/85, which is a 1x4 configuration.

Note: All 2(1x4) DSI serializers use the 2(1x4) map even when operating in single port mode (*Figure 159*).

ADI Confidential for New Revision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

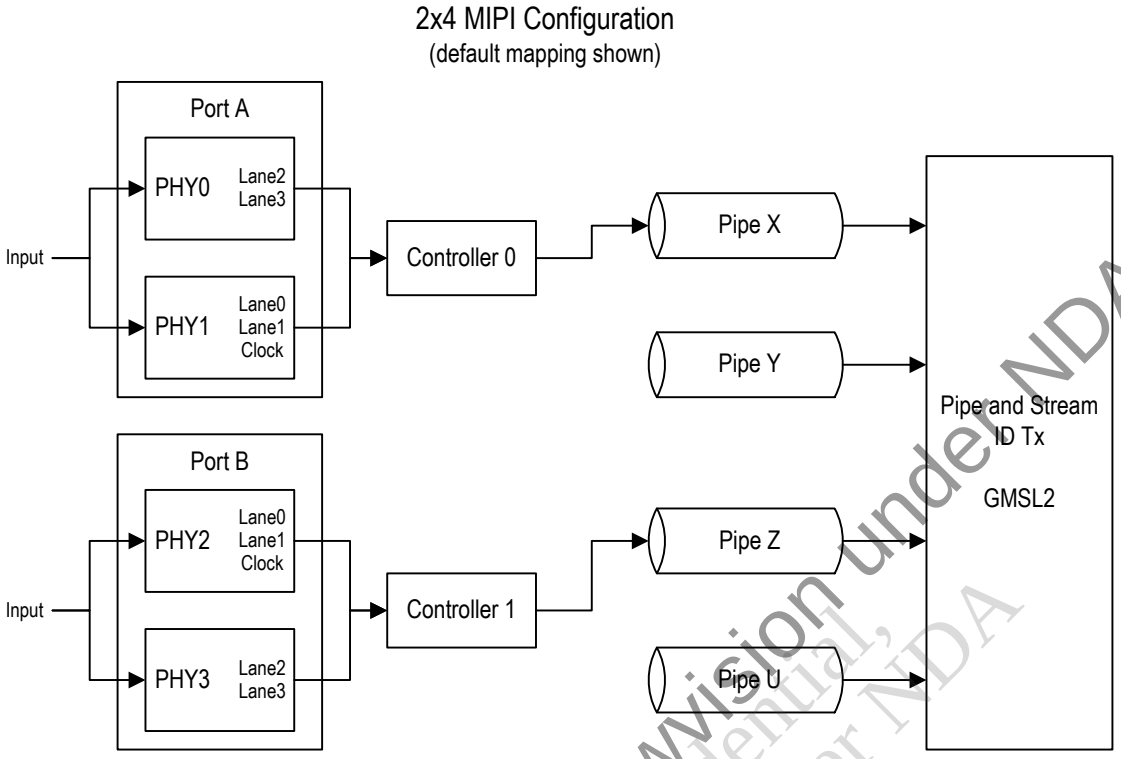


Figure 159. 2x4 Internal Mapping Diagram

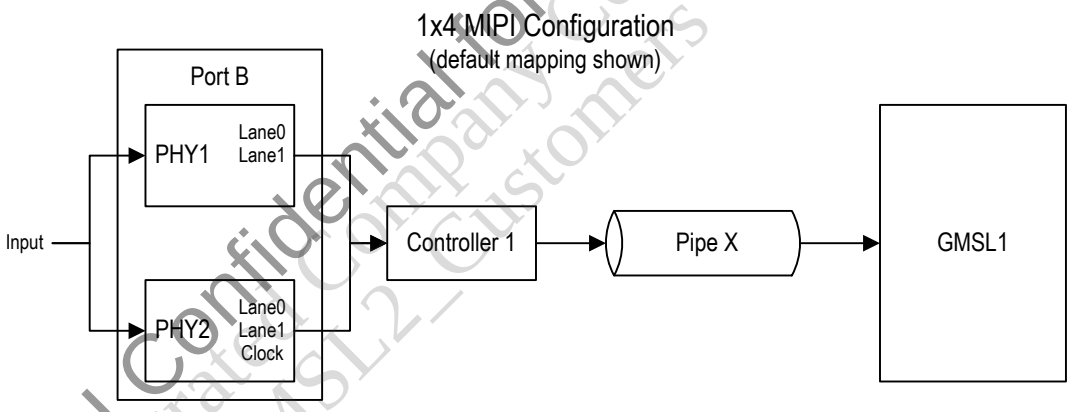


Figure 160. 1x4 Internal Mapping Diagram (MAX9283/85)

The 1x4 MIPI configuration requires that data pass through pipe X to meet GMSL1 backward compatibility requirements (Figure 160). This is done automatically with the default settings of the GMSL1-only DSI serializers, the MAX9283 and MAX9285.

GMSL2 DSI serializers have flexibility in mapping the data lanes and have configurable polarities for both data and clock lanes. The internal configuration changes depending on the MIPI configuration. Table 177 below summarizes the internal MIPI PHYs associated to each controller in each configuration.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Table 177. Internal MIPI PHY and Controller Mapping

MIPI Configuration	Part Numbers	DSI controller (ctrl0, ctrl1)	Internal MIPI PHY (master = clock)	Associated Data Lanes
2(1x4)	MAX96755/F MAX96757/F MAX96789 MAX96791	0	PHY1 (master)	Clock, D0, D1
			PHY0	D2, D3
		1	PHY2 (master)	Clock, D0, D1
			PHY3	D2, D3
1x4	MAX9283 MAX9285	1	PHY2 (master)	Clock, D0, D1
			PHY1	D2, D3

Each controller has two internal PHYs, and each PHY has two data lanes (for a total of four lanes per controller). The default mapping for a 2x4 configuration is shown in *Table 178*. Data lane arrangements are configurable.

Table 178. Default 2x4 Mapping

Port	Internal MIPI PHY (Data Lane)	Data Lane Mapping
Port A	PHY1 (D0)	Lane 0
	PHY1 (D1)	Lane 1
	PHY0 (D0)	Lane 2
	PHY0 (D1)	Lane 3
Port B	PHY1 (D0)	Lane 0
	PHY1 (D1)	Lane 1
	PHY0 (D0)	Lane 2
	PHY0 (D1)	Lane 3

The port Tx and controller Rx can be directly with lanes. Lane maps are controlled with registers below:

- 0x332[3:0]: phy0_lane_map[3:0] — PHY0 Lane Mapping
- 0x332[7:4]: phy1_lane_map[3:0] — PHY1 Lane Mapping
- 0x333[3:0]: phy2_lane_map[3:0] — PHY2 Lane Mapping
- 0x333[7:4]: phy3_lane_map[3:0] — PHY3 Lane Mapping

Example: A 2(1x4) configuration using port A and controller 0 with lane mapping (D3←→D0) will have a physical signal net connection of TxD3 to RxD0, and TxD0 to RxD3. To perform this lane map, write 0x72 (0b01110010) to 0x332. For a default configuration (no lane map), write 0x4E (0b01001110). Reference 'Port A' in Table 178 for the default mapping.

40.5.1.2 Lane Polarity

Data and clock lane polarities are referenced to a similar internal PHY and data lane structure as above. Use the registers below to invert the polarities.

- 0x334[2:0]: phy0_pol_map[2:0] — PHY0 Lane Polarity Setting
- 0x334[6:4]: phy1_pol_map[2:0] — PHY1 Lane Polarity Setting
- 0x335[2:0]: phy2_pol_map[2:0] — PHY2 Lane Polarity Setting
- 0x335[6:4]: phy3_pol_map[2:0] — PHY3 Lane Polarity Setting

Example: To invert the polarity of data lanes 3 and 0 from previous example, write 0x12 (0b010010) to 0x334.

ADI Confidential for New vision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

40.5.1.3 D-PHY Deskew

The MIPI D-PHY requires deskew calibration when operating above 1.5Gbps/lane. Each data lane supports clock-to-data lane deskew calibration. See the

MIPI D-PHY Deskew section for configuration details.

40.5.2 DSI Transmitter Settings

40.5.2.1 End-of-Transmission-packet (EoTp)

The MIPI D-PHY v1.2 and MIPI DSI v1.3 specifications incorporate an End of Transmission Packet (EoTp). The EoTp is an optional short packet appended to an HS transmission used to signal the end of the transmission. The EoTp detection can be disabled in the serializer to support legacy versions of MIPI. Ensure that both the transmitter (MIPI source) and receiver (GMSL2 DSI serializer) have EoTp enabled or disabled. The GMSL2 DSI serializers have EoTp reception enabled by default. To disable, clear the following bits:

- 0x389[4]: ctrl0_cfg_disable_eotp[0] — Disables EoTp packet support by Controller 0
- 0x399[4]: ctrl1_cfg_disable_eotp[0] — Disables EoTp packet support by Controller 1

40.5.2.2 Video Modes

GMSL2 DSI serializers accept either non-burst with sync pulses or non-burst with sync events video modes. In all cases, the video source must provide both horizontal and vertical sync information. By default, the GMSL2 DSI serializer expects non-burst with sync pulses; no further configuration is necessary if the source is in the same mode. If the source is sending non-burst with sync events, write 0x01 (0b01) to the controller:

- 0x380[7:6]: ctrl0_cfg_dpi_video_mode[1:0]
- 0x390[7:6]: ctrl1_cfg_dpi_video_mode[1:0]

The non-burst with sync events video mode also requires that the horizontal sync width and vertical sync width registers be configured. Horizontal parameters are programmed in number of internal PCLK cycles; vertical parameters are programmed in number of lines. Each parameter (i.e., HSW and VSW) are 12-bits in length, spread across multiple bytes. See the relevant registers below:

- **Controller 0 HS pulse width**
  - 0x385[7:0]: ctrl0_cfg_dpi_HS_width_l[7:0]
  - 0x387[3:0]: ctrl0_cfg_dpi_HS_width_h[3:0]
- **Controller 0 VS width**
  - 0x386[7:0]: ctrl0_cfg_dpi_VS_width_l[7:0]
  - 0x387[7:4]: ctrl0_cfg_dpi_VS_width_h[3:0]
- **Controller 1 HS pulse width**
  - 0x395[7:0]: ctrl1_cfg_dpi_HS_width_l[7:0]
  - 0x397[3:0]: ctrl1_cfg_dpi_HS_width_h[3:0]
- **Controller 1 VS width**
  - 0x396[7:0]: ctrl1_cfg_dpi_VS_width_l[7:0]
  - 0x397[7:4]: ctrl1_cfg_dpi_VS_width_h[3:0]

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribution Only to Maxim Integrated Company Confidential, Provided GMSL2_Customers Under NDA

Example: Port A HS width = 270 pixels, VS width = 300 lines, write 0x0E to 0x385, 0x2C to 0x386, and 0x11 to 0x387.

The internal PCLK is derived from MIPI clock using the following equation:

$$PCLK\ Frequency = \frac{MIPI\ Lane\ Rate\ \times \#\ of\ Lanes}{24}$$

40.5.2.3 Blanking Periods

The MIPI DSI protocol also incorporates different packet options comprising different data types (DT) to fill the blanking periods. The GMSL2 DSI serializers accept either blanking packets (DT = 0x19) or null packets (DT = 0x09) for this purpose. Blanking and null packets cannot be used together. To change the serializer mode to null packets (default is blanking packets), set the register bits below.

- 0x388[2]: ctrl0_cfg_dpi_use_null_pkt_bllp[0]
- 0x398[2]: ctrl1_cfg_dpi_use_null_pkt_bllp[0]

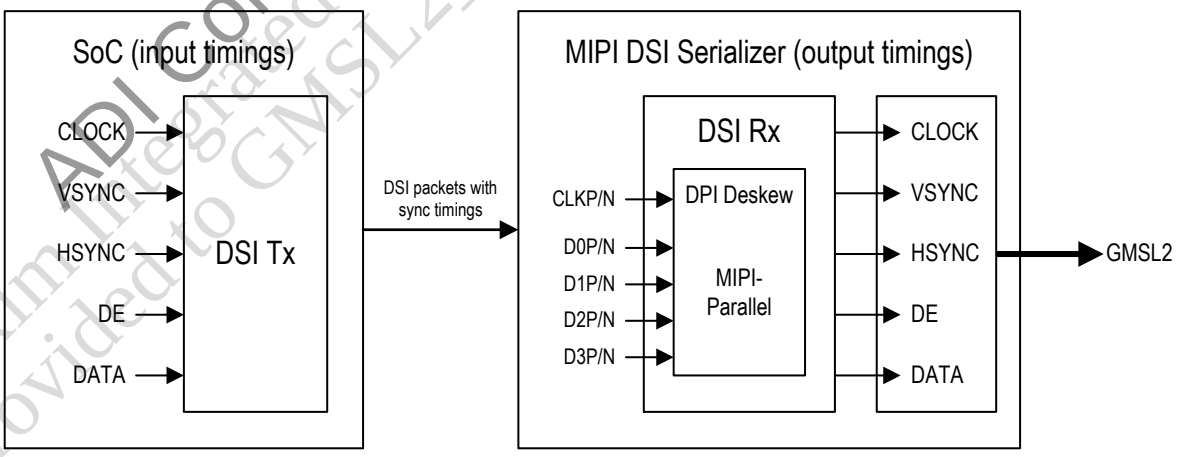
40.5.3 DPI Deskew

DPI Deskew is an internal function used to remove jitter and/or timing offset from the timing uncertainties introduced from the MIPI-to-parallel clock-domain crossings and Low Power (LP) to High Speed (HS) transitions. It requires programming the video timing parameters into the serializer registers.

These timing uncertainties can have ramifications when the video data is reconstructed by the deserializer. DPI Deskew is used to correct jitter that could result in incorrect output sync timings (horizontal) from the connected deserializer. Single display applications may not require deskew correction. Most displays can tolerate this amount of jitter; thus, DPI Deskew may be optional if there are no issues with the video output.

If using the asymmetric dual-view engine, however, DPI Deskew is necessary for a clean, asymmetrically split image. The function is required to accurately reconstruct the input timings of the MIPI DSI transmitter that are fed into the asymmetric engine to handle the splitting and allocation of the left and right asymmetric images.

See Figure 161 for a simplified block diagram of the MIPI DPI deskew conversion.



ADI Confidential for Revision under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA

Figure 161. MIPI DPI Deskew Conversion

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.5.3.1 DPI Deskew Requirements

For DPI Deskew to function properly, the following requirements must be met:

- The DSI source sends a constant horizontal line time (i.e., line-to-line and frame-to-frame).
- Each parameter within the horizontal line time (i.e., HFP, HBP, and HSW) is constant in absolute time.
- VS and HS signals must be active-high; use VS_Polarity and HS_Polarity bits to invert the signals if they are active-low at the input.
- The video timing parameters must be programmed into the DSI serializer and the MIPI DSI input must accurately follow the programmed video timing.

Program the following parameters into the serializer's registers (see Table 179). DPI Deskew is applied individually to each MIPI Port used. After all parameters are programmed, enable the DPI Deskew register.

- Horizontal Sync Width / Vertical Sync Width
- Horizontal Front Porch / Vertical Front Porch
- Horizontal Back Porch / Vertical Back Porch
- Horizontal Active Video / Vertical Active Video

Table 179. DPI Deskew Registers for Each MIPI Port

Parameter	MIPI Port A Register	MIPI Port B Register	Description
Horizontal Sync Width (HS)	(L) 0x385 [7:0] (H) 0x387 [3:0]	(L) 0x395 [7:0] (H) 0x397 [3:0]	12-bit field for horizontal sync pulse width, in number of PCLK cycles (pixels)
Vertical Sync Width (VS)	(L) 0x386 [7:0] (H) 0x387 [7:4]	(L) 0x396 [7:0] (H) 0x397 [7:4]	12-bit field for vertical sync pulse width, in number of lines (lines)
Vertical Front Porch (VFP)	(L) 0x3A5 [7:0] (H) 0x3A6 [3:0]	(L) 0x3B1 [7:0] (H) 0x3B2 [3:0]	12-bit field for vertical front porch, in number of lines (lines)
Vertical Back Porch (VBP)	(L) 0x3A6 [7:4] (H) 0x3A7 [7:0]	(L) 0x3B2 [7:4] (H) 0x3B3 [7:0]	12-bit field for vertical back porch, in number of lines (lines)
Vertical Active Video (VA)	(L) 0x3A8 [7:0] (H) 0x3A9 [3:0]	(L) 0x3B4 [7:0] (H) 0x3B5 [3:0]	12-bit field for vertical active video, in number of lines (lines)
Horizontal Front Porch (HFP)	(L) 0x3AA [7:0] (H) 0x3AB [3:0]	(L) 0x3B6 [7:0] (H) 0x3B7 [3:0]	12-bit field for horizontal front porch, in number of PCLK cycles (pixels)
Horizontal Back Porch (HBP)	(L) 0x3AB [7:4] (H) 0x3AC [7:0]	(L) 0x3B7 [7:4] (H) 0x3B8 [7:0]	12-bit field for horizontal back porch, in number of PCLK cycles (pixels)
Horizontal Active Video (HA)	(L) 0x3AD [7:0] (H) 0x3AE [4:0]	(L) 0x3B9 [7:0] (H) 0x3BA [4:0]	13-bit field for horizontal active video, in number of PCLK cycles (pixels)
DPI Deskew FIFO size	0x3A4 [7:6]	0x3B0 [7:6]	DPI Deskew FIFO size. Select larger FIFO size if input has larger skew.
DPI Deskew Enable	0x3A4 [0]	0x3B0 [0]	Enable DPI Deskew for respective MIPI controller

The GMSL2 GUI provides a DPI Deskew calculator tool that automatically generates a register .csv file containing all register writes needed for DPI Deskew.

An example of register writes for the following timing parameters can be seen below. This example uses MIPI Port A and the timing parameters in Table 180.

ADI Confidential for review under NDA
Maxim Integrated Confidential, provided to GMSL Customers Under NDA

Table 180. Example Timing Parameters for DPI Deskew

Parameter	Value
HS	42 (pixels)
VS	24 (lines)
VFP	12 (lines)
VBP	8 (lines)
VA	480 (lines)
HFP	28 (pixels)
HBP	36 (pixels)
HA	800 (pixels)

```
# Example DPI Deskew script using values in table above
# HS_WIDTH_L
0x80,0x385,0x2A
# VS_WIDTH_L
0x80,0x386,0x18
# HS_WIDTH_H/VS_WIDTH_H
0x80,0x387,0x00
# VFP_L
0x80,0x3A5,0x0C
# VBP_H
0x80,0x3A7,0x00
# VFP_H/VBP_L
0x80,0x3A6,0x80
# VRES_L
0x80,0x3A8,0xE0
# VRES_H
0x80,0x3A9,0x01
# HFP_L
0x80,0x3AA,0x1C
# HBP_H
0x80,0x3AC,0x02
# HFP_H/HBP_L
0x80,0x3AB,0x40
# HRES_L
0x80,0x3AD,0x20
# HRES_H
0x80,0x3AE,0x03
# FIFO/DESKEW_EN
0x80,0x3A4,0xC1
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.5.4 Dual-View Applications

Note: This section applies only to asymmetric or symmetric *Dual-View* applications. If dual-view is not used, skip to the *GMSL1 Settings* section below. If any of the GMSL PHYs is in GMSL1 mode and used for a dual-view application, the SSPLL registers must be programmed. Review the *GMSL1 SSPLL* section below.

The dual-view engines are accessible by routing the incoming video data to either pipe X or pipe Z. After passing through any dual-view engine, the incoming video is split into two separate internal video pipes. If the video is received by MIPI Port A, the video is processed by the dual-view block in pipe X the left video is split to pipe X (default), and the right half video (display B) is split to pipe Y. If received in pipe Z, the video splits to pipe Z and pipe U. See Figure 162 below.

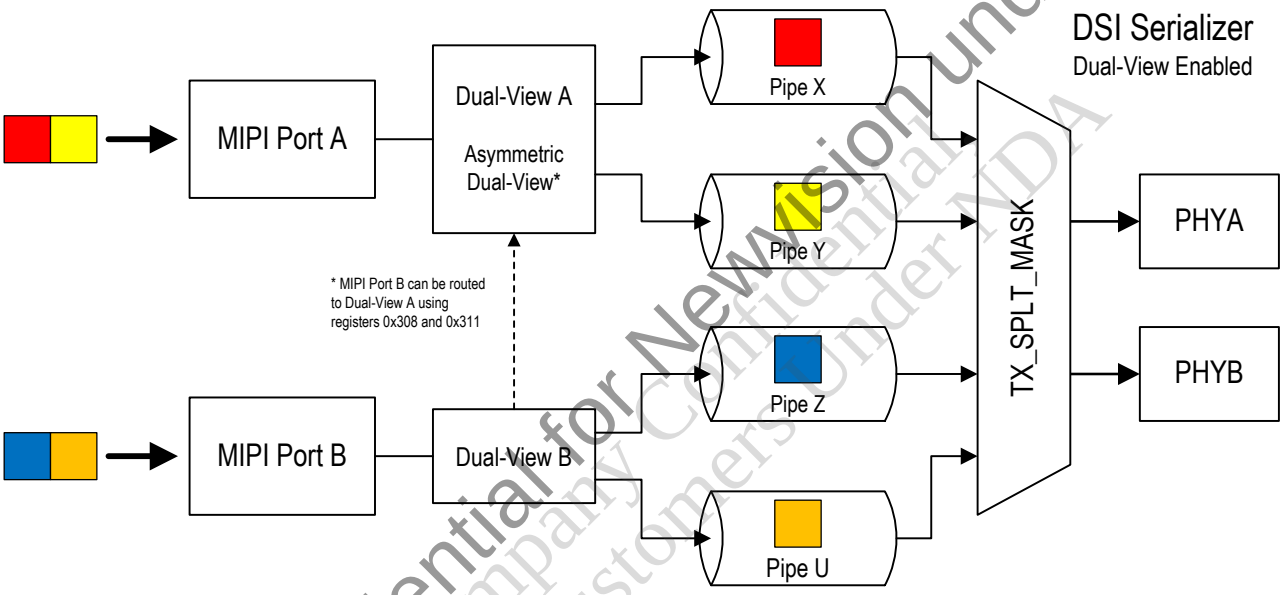


Figure 162. Dual-View Block Video Splitting

ADI Confidential for New Vision under NDA
Analog Devices Inc. Maxim Integrated Company Confidential Provided to GMSL2_Customers Under NDA

40.5.4.1 Asymmetric Dual-View

The MAX96789/MAX96791 support asymmetric and symmetric dual-view modes for both GMSL1 and GMSL2 combinations. For asymmetric dual-view, the DSI video source must provide a superframe comprising left and right videos of different resolutions. The right video will have uniformly padded dummy lines embedded in the video.

The asymmetric dual-view engine and all requirements are covered in detail in the Dual-View section.

Programming the asymmetric dual-view registers and parameters for each display (i.e., left and right) requires complex calculations to correctly regenerate the native timings across each link. The GMSL2 Asymmetric DSI GUI tool calculates the values to be programmed into the asymmetric dual-view registers.

The asymmetric dual-view engine is available only in pipe X, and incoming data must be routed to pipe X in order to use the asymmetric dual-view engine. For splitting applications, however, pipe Y must also be enabled. The following registers are used to enable additional video pipes:

- 0x02[5]: VID_TX_EN_Y[0] — Enable Video Transmit to Pipe Y
- 0x02[7]: VID_TX_EN_U[0] — Enable Video Transmit to Pipe U

Key timing registers asymmetric dual-view are provided below (using input timings):

- 0x700 to 0x718 (Incoming superframe timings)
- 0x720 to 0x740 (Timings for Display A)
- 0x750 to 0x770 (Timings for Display B)

The enable bits for the asymmetric dual-view engine are in register 0x719:

- 0x719[0]: ASYM_START[0]
- 0x719[1]: ASYM_EN_A[0]
- 0x719[2]: ASYM_EN_B[0]

The asymmetric dual-view engine supports the vertical blanking ratios in Table 181.

Table 181. Vertical Blanking Ratio Register

0x719[7:4]: LUT_TEMPLATE_SEL_B[3:0]	0x0: Use Look Up Table 0x1: 1:1 0x2: 2:1 0x3: 3:1 0x4: 4:1 0x5: 3:2 0x6: 4:3 0x7: 5:2 0x8: 5:3 0x9: 8:5 0xA: 2:1 inverted

ADI Confidential for New Vision under NDA
Analog Devices Confidential Distribute on Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.5.4.2 Symmetric Dual-View

The MAX96755/57 support only symmetric dual-view mode for 2x GMSL2 links. Note that the symmetric dual-view engine is separate from the asymmetric dual-view engine. There is a dedicated dual-view block for each MIPI Port (although typically only one is used at a time).

Dual-view on MIPI Port A, Pipe X:

- 0x32A[0]: DV_EN[0] — Enable the Dual-View Processor
- 0x32A[1]: DV_SPL[0] — Enable Dual-View Split Mode
- 0x32A[2]: DV_CONV[0] — Convert Side-by-Side Input to Pixel Interleaved
- 0x32A[5]: LINE_ALT[0] — Enable Line Alternating Mode

Dual-view on MIPI Port B, Pipe Z:

- 0x32D[0]: DV_EN[0] — Enable the Dual-View Processor
- 0x32D[1]: DV_SPL[0] — Enable Dual-View Split Mode
- 0x32D[2]: DV_CONV[0] — Convert Side-by-Side Input to Pixel Interleaved
- 0x32D[5]: LINE_ALT[0] — Enable Line Alternating Mode

The right image (display B) will be split into a separate video pipe (*Figure 156*). The video pipe enable registers (see above) will also need to be enabled. Note that it is possible to route video from MIPI Port B to the dual-view block on pipe X; see *Additional GMSL2 Register Configuration* for more information.

Example: For a symmetric dual-view application with a side-by-side video input at MIPI Port A, write 0x7 to 0x32A.

Refer to the Dual-View section for further information.

ADI Confidential for review under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.5.5 GMSL1 SSPLL

Note: This section is relevant to applications using asymmetric or symmetric *Dual-View* in GMSL1 mode.

SSPLL configuration is required for GMSL1 dual-view applications (applicable only to MAX96789/91). The GMSL link rate is dependent on the input PCLK frequency, and registers must be programmed to regenerate the native timings on each link.

Each GMSL PHY (SIOA or SIOB) is programmed separately. If operating in mixed mode (i.e., SIOA = GMSL2 and SIOB = GMSL1), only PHY B needs to be programmed. The GMSL2 DSI GUI tool calculates the values to be programmed into the SSPLL registers.

GMSL PHY A registers:

- 0x56E[2:0]: SSPLL_ODIV_EXP[2:0]
- 0x56E[7:3]: SSPLL_INDIV[4:0]
- 0x56F[7:0]: SSPLL_FB_DIV_L[7:0]
- 0x570[7]: SSPLL_FB_DIV_H[0]
- 0x570[6]: SSPLL_FORCE_DIV[0]
- 0x55B[7:0]: CSI_SSPLL_FB_FRACTION_IN_L[7:0]
- 0x55C[3:0]: CSI_SSPLL_FB_FRACTION_IN_H[3:0]
- 0x55C[7:5]: SSPLL_SEL_BCLK_A[2:0]

For GMSL PHY B:

- 0x571[2:0]: SSPLLB_ODIV_EXP[2:0]
- 0x571[7:3]: SSPLLB_INDIV[4:0]
- 0x572[7:0]: SSPLLB_FB_DIV_L[7:0]
- 0x573[7]: SSPLLB_FB_DIV_H[0]
- 0x573[6]: SSPLLB_FORCE_DIV[0]
- 0x56C[7:0]: CSI_SSPLLB_FB_FRACTION_IN_L[7:0]
- 0x56D[3:0]: CSI_SSPLLB_FB_FRACTION_IN_H[3:0]
- 0x55C[7:5]: SSPLL_SEL_BCLK_B[2:0]

Note: SSPLL_SEL_BCLK_A/B should not be put into AUTO when splitter mode is used; set SSPLL_SEL_BCLK_A/B[2] = 0b0.

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.5.6 GMSL1 Settings

The serializer and deserializer must have matching settings for normal operation and communication. Ensure that the BWS/HIBW, DRS, DBL, and HIM configurations match on both ends of the link.

DSI serializers accept only 24-bit RGB888. This is supported by BWS = 1 or HIBW = 1.

For GMSL PHY A:

- 0x407[5]: BWS[0]
- 0x407[6]: HIBW[0]
- 0x407[3]: DRS[0]
- 0x407[7]: DBL[0]
- 0x44D[7]: HIGHIMM[0]

For GMSL PHY B (available only in MAX96789/MAX96791):

- 0x607[5]: BWS[0]
- 0x607[6]: HIBW[0]
- 0x607[3]: DRS[0]
- 0x607[7]: DBL[0]
- 0x64D[7]: HIGHIMM[0]

The VESA or OLDI bit mapping selection can be chosen using registers:

- 0x415[1]: SEL_VESA[0] — PHY A
- 0x615[1]: SEL_VESA[0] — PHY B

If dual-view is used, SSPLL registers must be programmed. The configuration link is available to program remote GMSL1 deserializers without a valid input MIPI clock driving the link.

- 0x404[6]: CLINKEN[0] — PHY A
- 0x604[6]: CLINKEN[0] — PHY B

ADI Confidential for New vision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

40.5.7 Additional GMSL2 Register Configuration

This section outlines commonly used registers to route and configure video pipes, stream IDs, and other serializer registers. Note that the connected GMSL2 deserializer may require other register writes to receive video correctly.

Default settings work for most basic applications (e.g., single display).

Each video pipe (X, Y, Z, U) can select an incoming clock from a MIPI port. By default, Port A sends data to Pipe X, but it is possible to route Port B data to Pipe X (to satisfy some requirements such as GMSL1 data entering Pipe X or for dual-view processing) by using the below register. The MIPI Port enable is in the same register.

- 0x308: CLK_SELX/Y/Z/U
- 0x308: START_PORTA/B

Use the following registers to route the incoming data to the video pipe referenced for each MIPI Port. Both the clock and data should be routed to the desired video pipes using registers 0x308 and 0x311.

- 0x311: START_PORTAX/Y/Z/U
- 0x311: START_PORTBX/Y/Z/U

The MIPI Rx PHY configuration is controlled by phy_config[2:0]. All DSI serializers are capable of 2x4 except MAX9283, which is 1x4. Unused MIPI ports can be disabled to save power.

- 0x330[2:0]: phy_config[2:0]

The MIPI receiver can be reset by using the register below. The transmitter data should already be in a STOP state; after reset, the transmitter should send the START pattern.

- 0x330[3]: mipi_rx_reset[0]

Each video pipeline transmits video data with a tagged stream ID. The stream ID can be used as a filter on the deserializer side to determine what video data to receive. For example, it is possible to send two video streams of different IDs through the same GMSL PHY; the deserializer can be configured to then search for only a specific stream ID and output the video data tagged by that stream ID. This eliminates problems caused by mismatched physical cables because the video data is sent to both GMSL PHYs and is subsequently filtered by the deserializer for a specific stream ID.

- 0x53[1:0]: TX_STR_SEL[1:0] — Pipe X
- 0x57[1:0]: TX_STR_SEL[1:0] — Pipe Y
- 0x5B[1:0]: TX_STR_SEL[1:0] — Pipe Z
- 0x5F[1:0]: TX_STR_SEL[1:0] — Pipe U

Analog Confidential – Do not distribute outside Maxim Integrated Company
Provided to GMSL2 Customers Under NDA
ADI Confidential for New Version Under NDA

In splitter mode, video pipe output can be masked from the GMSL PHYs. This function is used for video duplication. For example, to send the video data in Pipe X to both GMSL PHY A and PHY B, use register 0x53 below.

- 0x53: TX_SPLT_MASK_A/B — Pipe X
- 0x57: TX_SPLT_MASK_A/B — Pipe Y
- 0x5B: TX_SPLT_MASK_A/B — Pipe Z
- 0x5F: TX_SPLT_MASK_A/B — Pipe U

Asymmetric DSI serializers have an additional video pipe crossbar stage before data transmission through the GMSL PHY. This can be useful for swapping videos to different PHYs at the last video pipeline stage before transmission.

- 0x326: CROSS_X/Y/Z/U

The DSI Rx accepts only 24-bit RGB888. In the MIPI packet header, DSI defines the 24-bit RGB888 data type as 0x3E. If the GMSL2 DSI serializer is connected to a GMSL2 CSI-2 deserializer, the data type definition in the MIPI packet header must be overwritten from 0x3E to 0x24 to conform with the CSI-2 packet header definition for RGB888 video. In this case, the following registers must be written for any active video pipes. Note that this is not required for other GMSL deserializers (i.e., eDP/DP, OLDI, etc.) that do not use MIPI data types.

- 0x321[5:0]: soft_dtx[5:0] (set this as 0x24)
- 0x31C[7]: soft_dtx_en[0] (Pipe X) (set this as 0x98)

- 0x322[5:0]: soft_dty[5:0] (set this as 0x24)
- 0x31D[7]: soft_dty_en[0] (Pipe Y) (set this as 0x98)

- 0x323[5:0]: soft_dtz[5:0] (set this as 0x24)
- 0x31E[7]: soft_dtz_en[0] (Pipe Z) (set this as 0x98)

- 0x324[5:0]: soft_dtu[5:0] (set this as 0x24)
- 0x31F[7]: soft_dtu_en[0] (Pipe U) (set this as 0x98)

To enable splitter mode (i.e., two deserializers connected to one serializer), write 0b11 to the register LINK_CFG (see below). Splitter mode requires that both deserializers be active or system LOCK will be lost. The asymmetric DSI serializers have an improved splitter mode (i.e., “glitchless”) where system LOCK can be maintained even if one of the two deserializers is powered down.

AUTO_LINK should be cleared and RESET_ONESHOT should be written when setting splitter mode. To optimize link lock time, follow the procedure in *GMSL2 Link Lock Time Optimization*

- 0x10[1:0]: LINK_CFG[1:0]
- 0x10[5]: RESET_ONESHOT[0]

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to CMS Customers Under NDA

40.6 Application Register Examples

The following examples in Table 182 are provided in the following format:

- Device Address, Register Address, Register Data to be written.
- "#" represents a comment and describes conditions of each test/write.

```
# This writes 0x04 to register address 0x330, to device address 0x80 (serializer)
0x80, 0x330, 0x04
```

- 0x80 is the DSI Serializer device address.
- Some register writes are omitted due to default values being suitable for the application.
  - Default values change depending on the hardware configuration at power-up (by CFG pins).
  - GMSL1 cases assume powering up in GMSL1 mode with HIM enabled.
  - GMSL2 cases assume powering up in GMSL2 mode in COAX mode at 6Gbps.
  - Mixed cases (i.e., GMSL1 and 2) assume powering up in GMSL2 mode, then writing the individual GMSL PHY to GMSL1 mode.
- Ideally, in all cases listed below, DPI Deskew is used for each MIPI Port. This configuration is excluded for concision.
- There are many ways to manipulate videos through each MIPI Port and Pipe to achieve a specific application. Examples below illustrate one method.

Table 182. Application Example Registers

Application 1: Single Display

```
# Default values will work for this application, but basic registers are outlined here
# Using 2 Lane MIPI Port A as input, outputting to GMSL PHY A
# Set only Pipe X to receive from Port A, power down Port B (save power)
0x80, 0x308, 0x5E
# Set 2 lane on MIPI Port A
0x80, 0x331, 0x31
```

Application 2: Dual Independent Displays

```
# Both GMSL PHY's are GMSL1 mode--this is only possible with Asymmetric Serializer MAX96789
# MIPI Port A data must be routed to Pipe X to output to GMSL PHY A in GMSL1 mode
# MIPI Port B data must be routed to Pipe Y to output to GMSL PHY B in GMSL1 mode
# Video Pipe Final stage mapping (default value)
0x80, 0x326, 0xD8
# Enable both MIPI Port A and B
0x80, 0x330, 0x06
# Enable Video on X and Y
0x80, 0x02, 0x33
# Set GMSL1 mode
0x80, 0x04, 0x32
# Route Port B
0x80, 0x308, 0x7E
0x80, 0x311, 0x61
# GMSL PHY A has BWS = 1, PHY B has BWS = 0/HIBW = 1
0x80, 0x407, 0x20
0x80, 0x607, 0x40
# PHY B uses OLDI mapping (PHY A default VESA unchanged)
0x80, 0x615, 0x01
# software datatype to RGB888 for all 3 video pipes used (X, Y, and Z)
0x80, 0x31C, 0x98
0x80, 0x321, 0x24
0x80, 0x31D, 0x98
0x80, 0x322, 0x24
0x80, 0x31E, 0x98
0x80, 0x323, 0x24
```

Maxim Integrated Confidential, Distribution Only Under NDA
For Internal Use Only
Maxim Integrated Confidential
For GMSL2 Customers Under NDA

```
# DPI Deskew for MIPI Port A (optional but recommended if source is compatible)
0x80, 0x385, 0x…
…
0x80, 0x3A4, 0x…
# DPI Deskew for MIPI Port B (optional but recommended if source is compatible)
0x80, 0x395, 0x…
…
0x80, 0x3B0, 0x…
# Enable Splitter mode and reset one shot
0x80, 0x10, 0x23
```

Application 3: Duplicated Display

```
# Video data received on MIPI Port B, send out same data to both GMSL PHY A and B, GMSL2
mode
# Both deserializers receive from Stream ID = 0
# Enable MIPI Port B
0x80, 0x330, 0x05
# DPI Deskew for MIPI Port B (optional but recommended if source is compatible)
0x80, 0x395, 0x…
…
0x80, 0x3B0, 0x…
# Registers for soft_dtz
0x80, 0x31E, 0x98
0x80, 0x323, 0x24
# Configure Pipe Z to send stream ID = 0, Pipe X to send stream ID = 2 (swapping ID's,
since X default = 0)
# Configure Pipe Z to send videos across both GMSL PHY's using TX_SPLT_MASK_A/B
0x80, 0x53, 0x02
0x80, 0x5B, 0x30
# Enable Splitter mode and reset one shot
0x80, 0x10, 0x23
```

Application 4: Symmetric Dual-View

```
# (MAX96789/91/F only because of Mixed Mode) Mixed Mode GMSL-1 (PHY A) + GMSL2 (PHY B)
# GMSL1 PHY A data must pass Pipe X, GMSL2 PHY B can be any pipe – will use Pipe Y for this
# Video data received on MIPI Port B, 4 lanes
# Input video is a side-by-side symmetric superframe
# BWS = 1
0x80, 0x407, 0x20
# SSPLL Settings for GMSL PHY A
0x80, 0x56F, 0x…
…
0x80, 0x55B, 0x…
# Setup MIPI ports to enable MIPI Port B
0x80, 0x330, 0x05
# Set both Pipe X and Pipe Y to receive clock from MIPI Port B – disable Port A
0x80, 0x308, 0x63
# Start data X and Y from Port B
0x80, 0x311, 0x30
# Enable side-by-side input dual-view processing on Pipe X
0x80, 0x32A, 0x07
# Send video pipes 1:1 (default in GMSL2 mode is OK)
0x80, 0x326, 0xE4
# soft_dtx/y registers
0x80, 0x31C, 0x98
…
0x80, 0x322, 0x24
# Enable video on pipe Y
0x80, 0x02, 0x73
# Set Pipe Y to send stream ID = 1, and output only on PHY B
0x80, 0x57, 0x21
# Set GMSL Modes for each PHY
0x80, 0x04, 0xB2
# DPI Deskew for MIPI Port B (optional but recommended if source is compatible)
```

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential, Under NDA
MAX96789 to GMSL2_Customers

```
0x80, 0x395, 0x…
…
0x80, 0x3B0, 0x…
# Set Splitter Mode and reset one shot
0x80, 0x10, 0x23
```

Application 5: Asymmetric Dual-View

```
# This example includes all writes to showcase an approximate number of registers for a
GMSL1 asymmetric splitter application, including some which are already correct default
values
# (MAX96789/91/F only) GMSL1 PHY A and GMSL1 PHY B, receive video from MIPI Port A
# 3:2 Vertical blanking ratio
# Set HIBW = 1 for PHY A, BWS = 1 for PHY B
0x80, 0x407, 0x40
0x80, 0x607, 0x20
# RGB888 and VESA format for both
0x80, 0x415, 0x03
0x80, 0x615, 0x03
# 4 Lanes
0x80, 0x331, 0x33
# 1x4 MIPI Port A
0x80, 0x330, 0x04
# Set Pipe X and Y to receive clock from Port A
0x80, 0x308, 0x5C
# Set Pipe X and Y to receive data from Port A
0x80, 0x311, 0x03
# soft_dtx/dty
0x80, 0x31C, 0x98
0x80, 0x321, 0x24
0x80, 0x31D, 0x98
0x80, 0x322, 0x24
# enable video on Pipe Y
0x80, 0x02, 0x73
# SSPLL PHY A (numbers are calculated based example output resolution and timings)
0x80, 0x56E, 0x08
0x80, 0x56F, 0x20
0x80, 0x570, 0x40
0x80, 0x55B, 0x97
0x80, 0x55C, 0x01
# SSPLL PHY B
0x80, 0x571, 0x09
0x80, 0x572, 0x1C
0x80, 0x573, 0x40
0x80, 0x56C, 0x6F
0x80, 0x56D, 0x81
# Default CROSS_* register value
0x80, 0x326, 0xE4
# DPI Deskew registers for MIPI Port A, controller 0
0x80, 0x385, 0x19
0x80, 0x386, 0x09
0x80, 0x387, 0x00
0x80, 0x3A5, 0x0C
0x80, 0x3A7, 0x00
0x80, 0x3A6, 0xF0
0x80, 0x3A8, 0xD0
0x80, 0x3A9, 0x02
0x80, 0x3AA, 0x3C
0x80, 0x3AC, 0x03
0x80, 0x3AB, 0xC0
0x80, 0x3AD, 0xD0
0x80, 0x3AE, 0x07
0x80, 0x3A4, 0xC1
# Configure Asymmetric Dual-View Registers
0x80, 0x700, 0x4C
```

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Company Confidential, Customers Under NDA
Maxim Integrated Confidential, Distribute only under NDA

```
0x80, 0x701, 0xF0
0x80, 0x702, 0x0E
0x80, 0x703, 0x70
0x80, 0x704, 0x89
0x80, 0x705, 0x06
0x80, 0x706, 0x74
0x80, 0x707, 0xBE
0x80, 0x708, 0x18
0x80, 0x709, 0x00
0x80, 0x70A, 0x00
0x80, 0x70B, 0xB0
0x80, 0x70C, 0x04
0x80, 0x70D, 0xD0C
0x80, 0x70E, 0x02
0x80, 0x70F, 0xB0
0x80, 0x710, 0x04
0x80, 0x711, 0xD0
0x80, 0x712, 0x07
0x80, 0x713, 0xD0
0x80, 0x714, 0x02
0x80, 0x715, 0x00
0x80, 0x716, 0x10
0x80, 0x717, 0x00
0x80, 0x718, 0x10
# ASYM_READ_REGS_A
0x80, 0x720, 0xE4
0x80, 0x721, 0x76
0x80, 0x722, 0x0E
0x80, 0x723, 0x87
0x80, 0x724, 0x2D
0x80, 0x725, 0x00
0x80, 0x726, 0xE1
0x80, 0x727, 0x4B
0x80, 0x728, 0x00
0x80, 0x729, 0x00
0x80, 0x72A, 0x00
0x80, 0x72B, 0x00
0x80, 0x72C, 0x0F
0x80, 0x72D, 0x00
0x80, 0x72E, 0x00
0x80, 0x72F, 0x05
0x80, 0x730, 0xF4
0x80, 0x731, 0x02
0x80, 0x732, 0x00
0x80, 0x733, 0x00
0x80, 0x734, 0x00
0x80, 0x735, 0x37
0x80, 0x736, 0x00
0x80, 0x737, 0x00
0x80, 0x738, 0xB0
0x80, 0x739, 0x04
0x80, 0x73A, 0x5F
0x80, 0x73B, 0x00
0x80, 0x73C, 0xD0
0x80, 0x73D, 0x02
0x80, 0x73E, 0xE5
0x80, 0x73F, 0xB5
0x80, 0x740, 0x00
# ASYM_READ_REGS_B
0x80, 0x750, 0x50
0x80, 0x751, 0x54
0x80, 0x752, 0x06
0x80, 0x753, 0xEC
```

ADI Confidential for Newvision under NDA
Maxim Confidential Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

```
0x80, 0x754, 0x13
0x80, 0x755, 0x00
0x80, 0x756, 0x34
0x80, 0x757, 0x21
0x80, 0x758, 0x00
0x80, 0x759, 0x00
0x80, 0x75A, 0x00
0x80, 0x75B, 0x00
0x80, 0x75C, 0x0A
0x80, 0x75D, 0x00
0x80, 0x75E, 0x48
0x80, 0x75F, 0x03
0x80, 0x760, 0xF8
0x80, 0x761, 0x01
0x80, 0x762, 0x00
0x80, 0x763, 0x00
0x80, 0x764, 0x00
0x80, 0x765, 0x1E
0x80, 0x766, 0x00
0x80, 0x767, 0x00
0x80, 0x768, 0x20
0x80, 0x769, 0x03
0x80, 0x76A, 0x32
0x80, 0x76B, 0x00
0x80, 0x76C, 0xE0
0x80, 0x76D, 0x01
0x80, 0x76E, 0x92
0x80, 0x76F, 0x4F
0x80, 0x770, 0x00
#Select 3:2 vertical LUT pattern/ Enable displays A and B, start asym engine
0x80, 0x719, 0x57
# Enable Splitter mode
0x80, 0x10, 0x23
```

ADI Confidential for New vision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.7 Status and Debug Registers

The following steps and registers can be used to determine the source of a problem if there is unexpected behavior.

Ensure that the serializer is correctly configured:

- The number of programmed lanes matches the number of lanes of the DSI Source.
- MIPI Ports are enabled (phy_config).
- The video source is sending 24-bit RGB888 data.
- Correct serial link mode configured (i.e., GMSL1 or GMSL2 mode).
- Video pipe(s) correctly configured and enabled.
- EoTp settings are matched between the DSI source and MIPI D-PHY receiver.
- Timing registers (i.e., DPI Deskew) matches the source video timings.
    - The source device is compatible with the DPI Deskew function enabled by the DSI serializer.

Default values of registers may be different depending on the hardware configuration at power-up. CFGx pins will determine some register default, including the clock routing (0x308) or video pipe swaps (0x326).

SoC data and clock lanes should be matched 1:1 to the DSI serializer lanes unless lane mapping is used. Ensure that the lane mappings are correct (both in the schematic and in the registers):

- If lane mapping / polarity adjustments are required, make sure registers are correct.
- ECC errors may occur if the data lanes are mapped incorrectly.
- Check schematic symbol of the DSI serializer and SoC. Pin-to-lane assignments may be incorrectly drawn.

Confirm that the serializer is receiving a valid pixel clock. Read register PCLKDET to check if the serializer is receiving a valid MIPI clock in the correct video pipeline.

- Ensure that the MIPI clock lanes are routed correctly (e.g., no skew and/or length mismatch, +/- incorrectly routed to +/-, etc.).
- Clock is continuous clock mode.
- The proper start/stop pattern is being received by serializer.

For both GMSL1 and GMSL2 PCLKDET:

- 0x102/0x10A/0x112/0x11A: PCLKDET (X/Y/Z/U)

Video overflow registers can also be found in the above registers. These registers indicate that the incoming video is overflowing the video channel, and that either the video rate should be reduced or the link rate should be increased.

For GMSL1 PCLKDET:

- 0x1C9/0x20C/0x24F/0x292: PCLKDET (X/Y/Z/U)

DSI error registers indicate statuses of some error cases that may appear for a non-ideal system. There are separate DSI error registers for each controller. ECC errors (1-bit or 2-bit), CRC errors, and HS receiver timeouts can be detected using these registers. The ctrl*_dsi_err_h register shows the position of the ECC error if it is a 2-bit error.

ADI Confidential for New Vision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

- 0x3A0: ctrl0_dsi_err_l — Controller 0
- 0x3A1: ctrl0_dsi_err_h — Controller 0

- 0x3A2: ctrl1_dsi_err_l — Controller 1
- 0x3A3: ctrl1_dsi_err_h — Controller 1

Each video pipe has a sync detect status register. To monitor video sync signals, read the registers to ensure that the Horizontal Sync, Vertical Sync, and Data Enable signals are stable. Polarities of incoming HS/VS can also be found in these registers. The HS/VS/DE_DET registers will not be stable if DPI Deskew is enabled and there is a mismatch of source and receiver timing information.

- 0x055D[0]: HS_POL_X[0] — Pipe X
- 0x055D[1]: VS_POL_X[0] — Pipe X

- 0x055D[4]: HS_DET_X[0] — Pipe X
- 0x055D[5]: VS_DET_X[0] — Pipe X
- 0x055D[6]: DE_DET_X[0] — Pipe X

- 0x055E — Pipe Y
- 0x055F — Pipe Z
- 0x056A — Pipe U

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

40.8 Feature Availability by Device

Table 183 contains the features available to different devices in the GMSL2 DSI Serializer family.

Table 183. GMSL2 DSI Serializers Feature Availability

Device Number	GMSL PHY	MIPI Configuration	Dual-View Modes	HDCP	Max PCLK (MHz) per pipe
MAX9283	GMSL1	1x4	None	No	150
MAX9285	GMSL1	1x4	None	Yes	150
MAX96755	GMSL1, GMSL2, or 2x GMSL2	1x4 or 2x4	Symmetric	No	215
MAX96755F	GMSL1, GMSL2, or 2x GMSL2	1x4 or 2x4	Symmetric	No	160
MAX96755H	GMSL1, GMSL2, or 2x GMSL2	1x4 or 2x4	Symmetric	No	104
MAX96757	GMSL1, GMSL2, or 2x GMSL2	1x4 or 2x4	Symmetric	Yes	215
MAX96757F	GMSL1, GMSL2, or 2x GMSL2	1x4 or 2x4	Symmetric	Yes	160
MAX96757H	GMSL1, GMSL2, or 2x GMSL2	1x4 or 2x4	Symmetric	Yes	104
MAX96789	GMSL1, GMSL2, 2x GMSL1, 2x GMSL2, or GMSL1 and GMSL2	1x4 or 2x4	Symmetric and Asymmetric	No	215
MAX96789F	GMSL1, GMSL2, 2x GMSL1, 2x GMSL2, or GMSL1 and GMSL2	1x4 or 2x4	Symmetric and Asymmetric	No	160
MAX96789H	GMSL1, GMSL2, 2x GMSL1, 2x GMSL2, or GMSL1 and GMSL2	1x4 or 2x4	Symmetric and Asymmetric	No	104
MAX96791	GMSL1, GMSL2, 2x GMSL1, 2x GMSL2, or GMSL1 and GMSL2	1x4 or 2x4	Symmetric and Asymmetric	Yes	300
MAX96791F	GMSL1, GMSL2, 2x GMSL1, 2x GMSL2, or GMSL1 and GMSL2	1x4 or 2x4	Symmetric and Asymmetric	Yes	160
MAX96791H	GMSL1, GMSL2, 2x GMSL1, 2x GMSL2, or GMSL1 and GMSL2	1x4 or 2x4	Symmetric and Asymmetric	Yes	104

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

41 Dual OLDI

41.1 Overview

OpenLDI (OLDI) is a high-bandwidth digital-video interface that uses low-voltage differential signaling (LVDS) to transmit digital video without digital-to-analog conversion or complex decoding protocols. GMSL2 OLDI deserializers convert single- or dual-link GMSL2 input to single or dual OLDI video output with flexible output configurations, a color lookup table (LUT) for gamma correction, and programmable spread-spectrum clocking (SSC). Refer to Feature Availability by Device Family.

OLDI output configurations:

- Single port (4 or 8 lanes) or dual port (2x4 lanes)
- 1x4, 2x4, or 1x8 OLDI output lane configurations
- Each port supports pixel clock rates up to 150MHz (300MHz in dual port mode). Refer to the Interface-Specific Bandwidth Calculations section for additional information.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

41.2 Operation

GMSL2 OLDI deserializers receive GMSL data from the serial link and convert it to video data. The video data is then transmitted by the video pipes and mapped to ports A and/or B of the output interface. The output video interface has user-configurable port arrangements and can transmit single or dual OLDI video in either OLDI or VESA formats (see LVDS Data Mapping below).

Table 184 below contains the OLDI output video interface pin numbers, names, and descriptions.

Table 184. OLDI Interface Pin Descriptions

Pin Number	Pin Name	Function
30	TxOUT_A3+	LVDS Data Output Port A
31	TxOUT_A3-	LVDS Data Output Port A
32	TxOUT_A2+	LVDS Data Output Port A
33	TxOUT_A2-	LVDS Data Output Port A
36	TxOUT_A1+	LVDS Data Output Port A
37	TxOUT_A1-	LVDS Data Output Port A
38	TxOUT_A0+	LVDS Data Output Port A
39	TxOUT_A0-	LVDS Data Output Port A
34	TxCLK_OUTA+	LVDS Clock Output Port A
35	TxCLK_OUTA-	LVDS Clock Output Port A
46	TxOUT_B3+	LVDS Data Output Port B
47	TxOUT_B3-	LVDS Data Output Port B
48	TxOUT_B2+	LVDS Data Output Port B
49	TxOUT_B2-	LVDS Data Output Port B
52	TxOUT_B1+	LVDS Data Output Port B
53	TxOUT_B1-	LVDS Data Output Port B
54	TxOUT_B0+	LVDS Data Output Port B
55	TxOUT_B0-	LVDS Data Output Port B
50	TxCLK_OUTB+	LVDS Clock Output Port B
51	TxCLK_OUTB-	LVDS Clock Output Port B

ADI Confidential for NEW vision under NDA
Analog Devices Confidential only under NDA
Maxim Integrated Confidential,
Provided to GMSL2_ Customer Under NDA

The GMSL2 OLDI deserializer generates the OLDI clock output from the tracked and measured clock frequency of the video input at the connected serializer. Figure 163 shows the LVDS pulse positions with the OLDI clock.

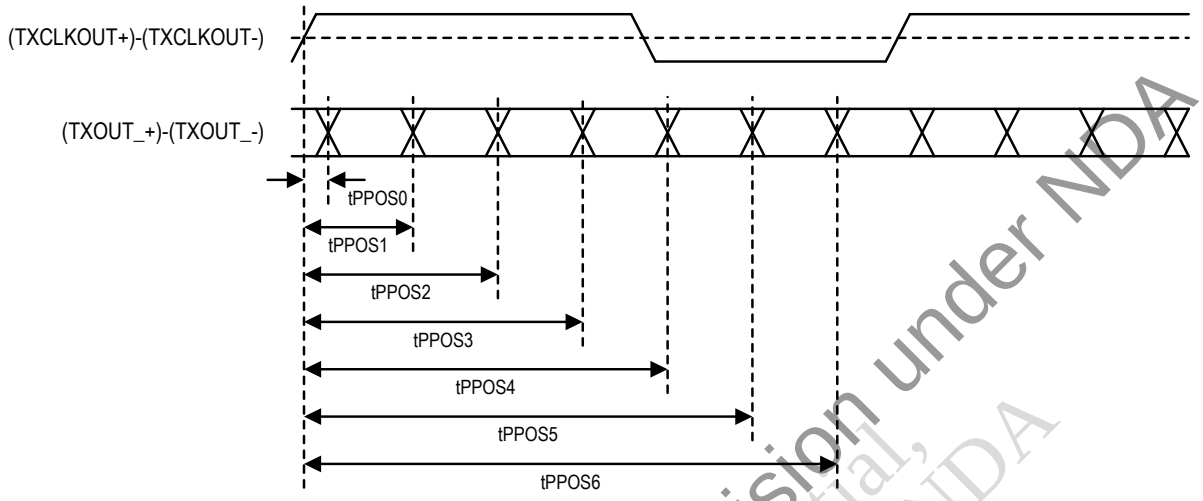


Figure 163. LVDS pulse positions

GMSL2 OLDI deserializers can be configured as single port (4 or 8 lanes) or dual port (2x4 lanes) to accommodate a range of display applications. Each port supports pixel clock (PCLK) rates up to 150MHz and a combined PCLK of 300MHz in dual-port mode. Video data can be mapped to OLDI ports A, B, or both. Additionally, video lanes can be inverted and/or swapped to assist in compatibility with display interfaces. Both OLDI and VESA output formats are supported.

ADI Confidential for Review under NDA
Analog Devices Confidential, Not Maxim Integrated Company Confidential, Provided to GMSL2_ Customer Under NDA

41.2.1 LVDS Data Mapping

The following sections provide the LVDS data mapping for single and dual-port OLDI output.

41.2.1.1 Single Pixel per Clock Output

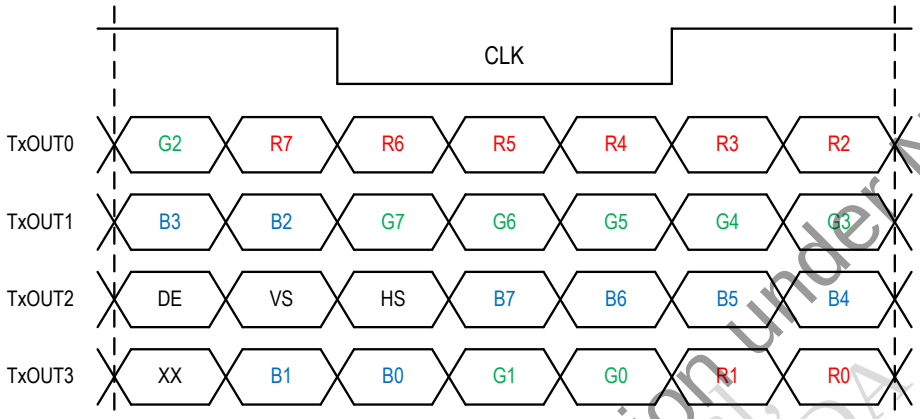


Figure 164. Single OLDI Port Mapping (OLDI/Format 1)

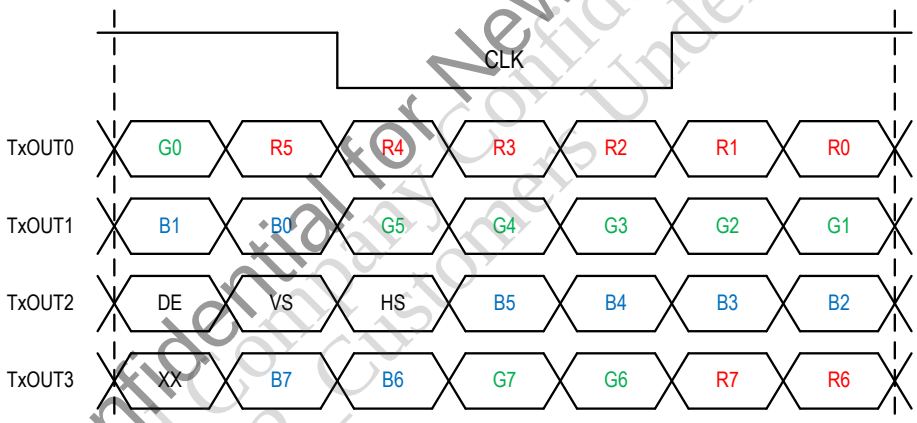


Figure 165. Single OLDI Port Mapping (VESA/Format 2)

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

41.2.1.2 Dual Pixel per Clock Output

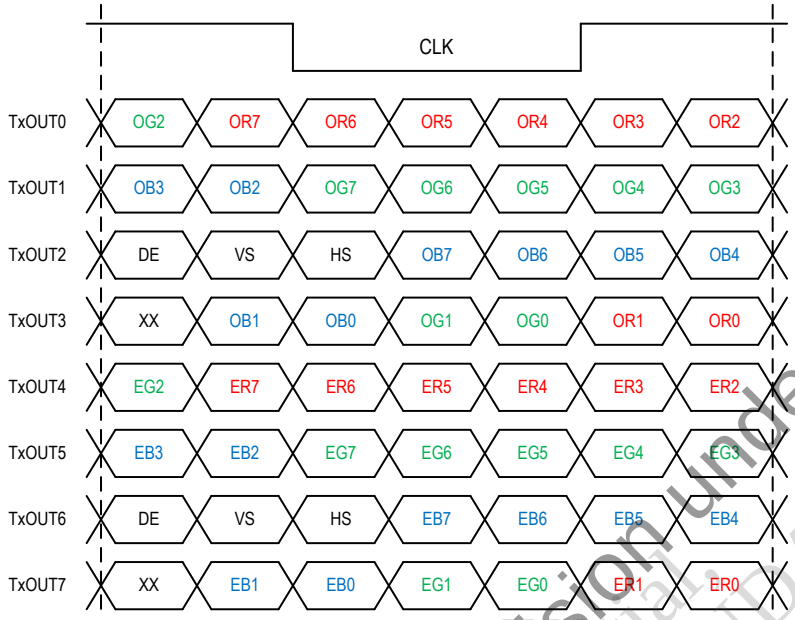


Figure 166. Dual OLDI Port Mapping (OLDI/Format 1)

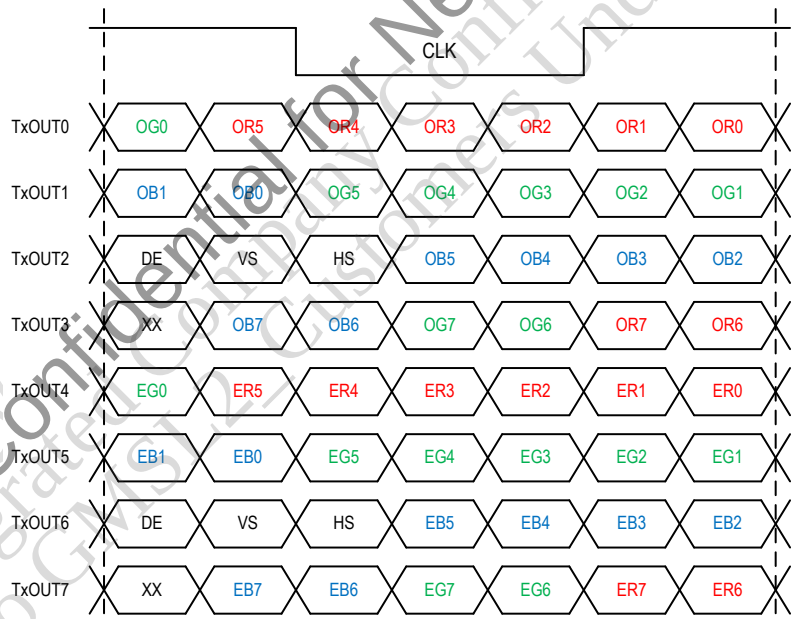


Figure 167. Dual OLDI Port Mapping (VESA/Format 2)

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

41.2.2 Output Programming

- Output ports are configurable as 1x4, 1x8, or 2x4 lanes. Refer to the Configuration section for programming details related to this block.
- OLDI ports can be configured as 4-lane (for RGB888) or 3-lane (for RGB666) with the OLDI_4TH_LANE register bit.
- The OLDI output lanes can be inverted and/or swapped. The LANE_SEL_* registers select the output pin mapping for the port and lane designated by the register name; the LANE_INV_* registers invert the polarity of the port and lane designated by the register name.
- Vertical sync (VS) can be viewed on GPIO01. An optional single-cycle glitch filter is enabled with the VS_OUT_EN register bit.

41.2.3 Color Lookup Table (LUT)

The LUT enables 1:1 translation of 8-bit RGB color input data to any 8-bit RGB color output values. The feature is primarily intended for color filtering and gamma correction. The LUT consists of three discrete color channels that accept 8-bit wide entry for each color (i.e., red, green, and blue) with 256-entry depth. Refer to the Color Lookup Table (LUT) section for additional information and configuration details.

41.2.4 Spread-Spectrum Clocking

The OLDI output port can be programmed to have 0.5% center spread-spectrum clocking (SSC) with a 20kHz to 40kHz triangular modulation frequency. It is possible to program the spread-spectrum percentage up to 4% with the config_spread_bit_ratio bitfield (register DPLL_3).

Note: OLDI SSC mode programming is independent of the GMSL high-speed clock SSC mode. Reference the *Spread-Spectrum Clocking (SSC)* section for GMSL high-speed clock SSC information.

41.2.5 Odd/Even Pixel Interleaving in Dual OLDI Mode

The OLDI output processor can split the output of the two OLDI ports at half the PCLK frequency by using the active-high or active-low versions of the DE or HS signals for even/odd pixel alignment or no alignment.

In single-port mode, the maximum OLDI PCLK of 150MHz supports up to FHD (1920x1080 24-bit/60Hz) or equivalent. In dual-port mode, the interleaved even and odd pixels of the video stream are split and routed to the A and B output ports and transmitted to a single display. This process divides the video PCLK by a factor of two at the OLDI ports. For FHD video, the clocking on each port is approximately 75MHz and is below the FM radio band.

In dual-port mode, the GMSL2 OLDI deserializers support up to a 300MHz PCLK if both GMSL2 link inputs are used. This enables display of interleaved WQXGA (2560x1600 24-bit/60Hz) or equivalent video. If only one GMSL link is used, the maximum PCLK is limited to approximately 217MHz by the 6Gbps link bandwidth. This enables display of 2560x1080 24-bit/60Hz video or equivalent.

Analog Devices Confidential, Distribution Limited
Maxim Integrated Company Confidential, Customers Under NDA
Provided to GMSL2 Customers Under NDA
ADI Confidential for Newvision under NDA

41.2.6 Video Replication and Dual-View Splitter Mode

OLDI deserializer replication enables identical FHD (or equivalent) video to be sent to two displays. Dual-view splitter mode supports column-interleaved FHD displays with different content split from a single video stream. Ports A and B can be swapped as needed. Note that side-by-side to column-interleaved conversion is processed on the serializer side; reference the Dual-View section for additional information.

41.3 Application Examples

GMSL2 OLDI deserializers have a single video pipeline and support single video streams, video replication, and dual-view splitting.

The application examples illustrated below show a GMSL2 system with an HMDI source and HDMI serializer. These configurations are also possible with MIPI DSI and CSI-2 sources. Reference the appropriate serializer section for configuration details. GMSL2 OLDI deserializer programming is detailed below in the Configuration section.

In Figure 168, the serializer transmits a single video stream to the OLDI deserializer. The OLDI deserializer outputs to one connected display in either single or dual OLDI mode.

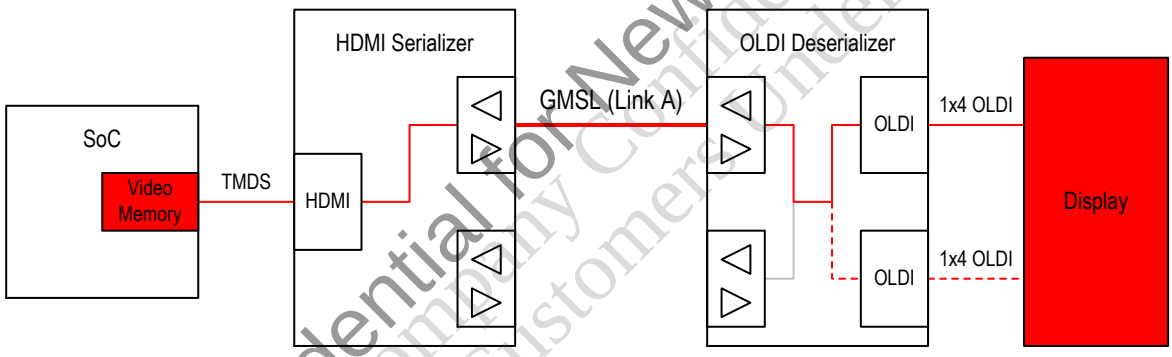


Figure 168. Serializer Single Link Mode with OLDI Deserializer Single or Dual Mode

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Subject to Mutual NDA
Maxim Integrated Company Confidential, Customer Under NDA
Provided to GMSL2_

In Figure 169, the serializer in pixel-interleaved dual-view mode receives a superframe from the video source. The dual-view block splits the superframe and transmits the pixel-interleaved data on the serial link. The OLDI deserializer operates in splitter mode to split pixel-interleaved data into two video streams. The streams are transmitted from separate OLDI ports to two connected displays.

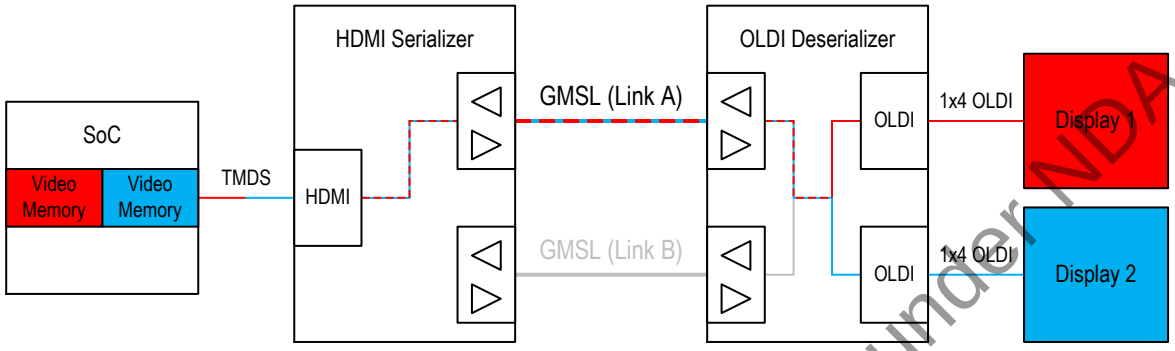


Figure 169. Serializer Dual-View Mode with OLDI Deserializer Splitter Mode

The GMSL2 system in *Figure 170* is configured for GMSL2 dual link mode. Video is transmitted from the serializer on dual GMSL2 links. The OLDI deserializer outputs to one connected display in either single or dual OLDI mode.

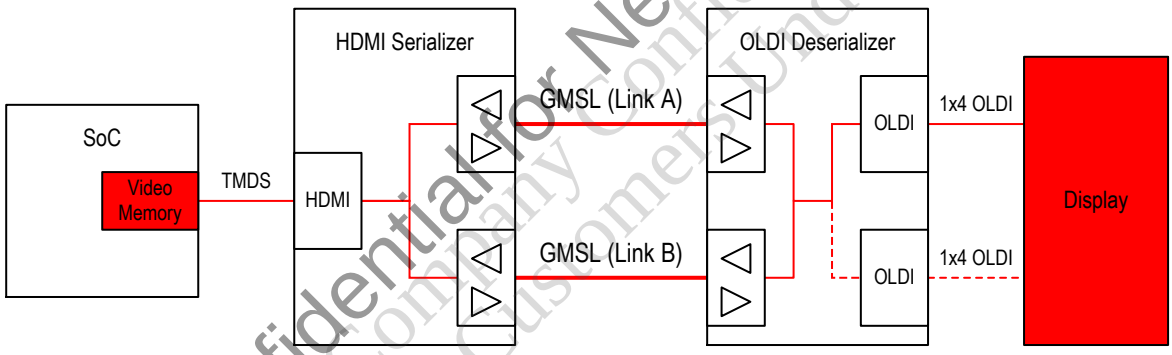


Figure 170. Single Video Stream and GMSL2 Dual Link Mode

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distributed Under NDA
Maxim Integrated Company Confidential, Customers Under NDA
Provided to GMSL2_Customers

The serializer in Figure 171 transmits a replicated video stream in GMSL2 dual link mode to two connected OLDI deserializers. Each OLDI deserializer outputs to one connected display in either single or dual OLDI mode.

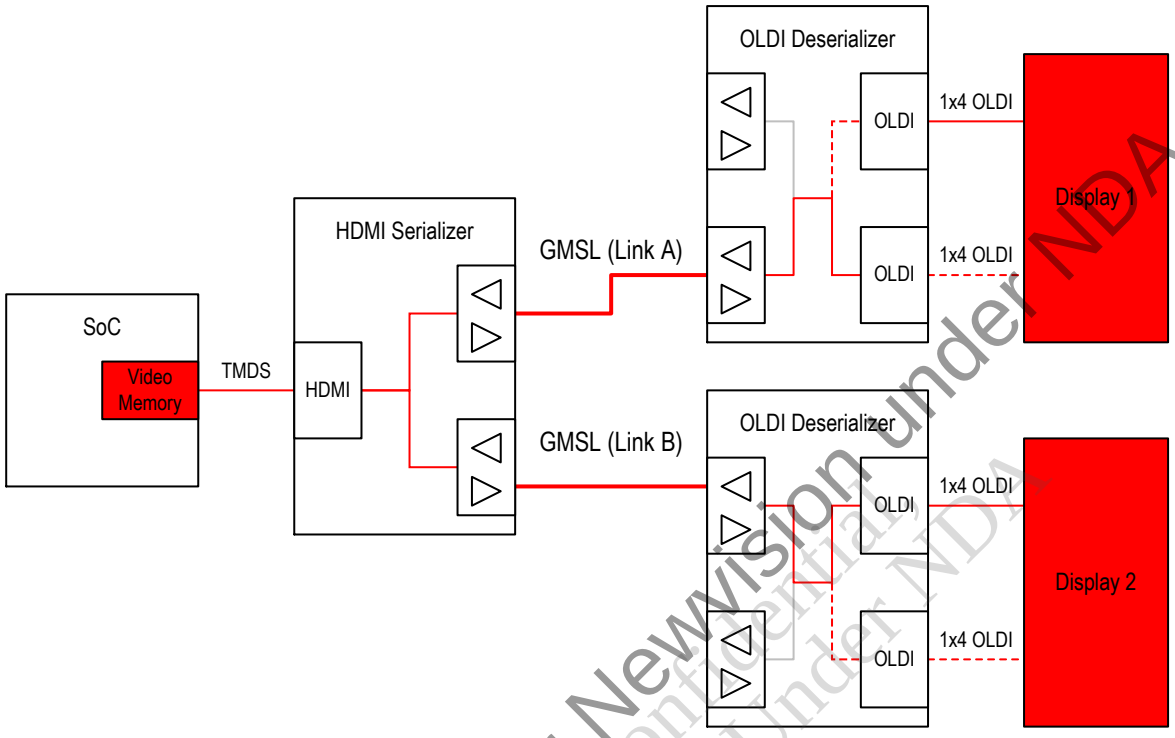


Figure 171. Single Video Stream with Serializer Replication

In Figure 172, the serializer transmits one video stream to the OLDI deserializer. The deserializer replicates the video by sending identical streams to both OLDI ports, which are connected to separate displays.

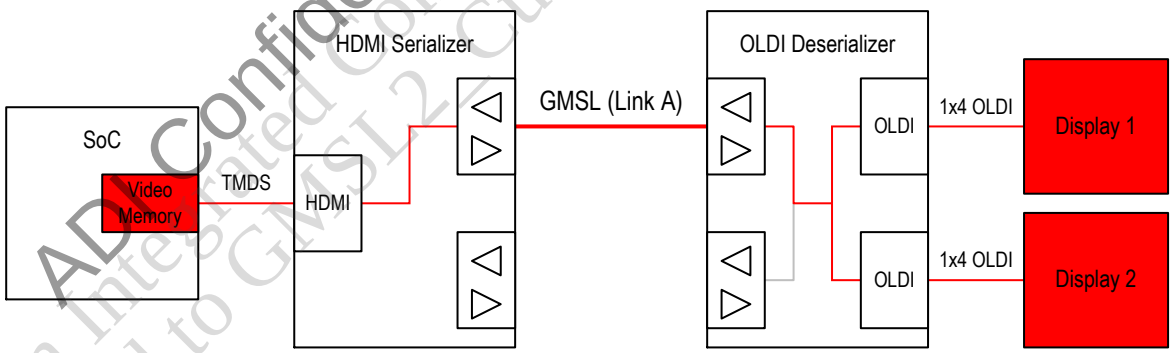


Figure 172. Single Video Stream with OLDI Deserializer Replication

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL 2 Customers Under NDA

41.4 Configuration

This section contains the registers relevant to OLDI output. OLDI port configuration registers are presented in Table 185. Reference the Programming Examples section below for use-case applications. Note that applications may require serializer configuration (refer to appropriate sections in this document for details).

For proper operation in OLDI Splitter Mode, the display timing parameters must have an even number of PCLK cycles before the OLDI split. If this is not applied, the synchronization between line starts of the two LVDS ports may become skewed.

Program the output format of the LVDS port with OLDI_FORMAT with the format of the connected display to ensure proper color mapping.

Note: All registers in Table 185 (except OLDI SSEN) must only be set when VID_LOCK = 0. The OLDI_SPL_POL bit in register OLDI1 (0x1CE) must always be set to 0 for dual OLDI applications.

Table 185. OLDI Port Configuration Registers

Register Address	Bitfield	Description	Decode
0x1CE	OLDI_OUTSEL	Selects which OLDI port to use for this data output.	0b0: Port A 0b1: Port B
0x1CE	OLDI_FORMAT	Output format of LVDS port.	0b0: OLDI 0b1: VESA
0x1CE	OLDI_4TH_LANE	OLDI lane count.	0: 4 lanes 1: 3 lanes
0x1CE	OLDI_SWAP_AB	Swaps OLDI ports A and B.	0b0: Ports A and B not swapped 0b1: Ports A and B swapped
0x1CE	OLDI_SPL_EN	OLDI Splitter enable.	0b0: Splitter disabled 0b1: Splitter enabled
0x1CE	OLDI_SPL_MODE	OLDI Splitter Mode.	0b00: Random 0b01: Split with HS 0b10: Split with VS 0b11: Split with DE
0x1CE	OLDI_SPL_POL	Set to 0 for dual OLDI applications.	0b0: Falling edge 0b1: Rising edge
0x1CF	PD_LVDS_B	Selects power-up/power-down status for LVDS output driver B.	0b0: LVDS output driver B powered up 0b1: LVDS output driver B powered down
0x1CF	PD_LVDS_A	Selects power-up/power-down status for LVDS output driver A.	0b0: LVDS output driver A powered up 0b1: LVDS output driver A powered down
0x1CF	OLDI_DUP	Selects whether or not to duplicate OLDI at both ports	0b0: OLDI not duplicated 0b1: OLDI duplicated at both ports
0x1CF	SSEN	Enables spread spectrum OLDI output clock. Spread percentage is adjusted from OLDIPLL registers (reference the Spread-Spectrum Clocking (SSC) section). Default percentage is 0.5%.	0b0: Spread spectrum disabled 0b1: Spread spectrum enabled

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2 Customers

Table 186 contains registers used to output polarity inversion and output pin mapping. Polarity is inverted on a lane-by-lane basis. The pin mapping registers enable programming the pin assignments for the OLDI output ports and lanes.

Table 186. OLDI Output Polarity Inversion and Pin Mapping Registers

Register Address	Bitfield	Description	Decode
0x1D0	LANE_INV_B0	Lane B0 polarity inversion	0b0: Lane polarity not inverted 0b1: Lane polarity inverted
0x1D0	LANE_INV_A0	Lane A0 polarity inversion	0b0: Lane polarity not inverted 0b1: Lane polarity inverted
0x1D1	LANE_INV_B1	Lane B1 polarity inversion	0b0: Lane polarity not inverted 0b1: Lane polarity inverted
0x1D1	LANE_INV_A1	Lane A1 polarity inversion	0b0: Lane polarity not inverted 0b1: Lane polarity inverted
0x1D0	LANE_SEL_B0[2:0]	Selects OLDI Port B Lane 0 output pins	0b000: Output from B0 0b001: Output from B1 0b010: Output from B2 0b011: Output from B3 0b100: Output from BCLK 0b101: Reserved 0b110: Reserved 0b111: Reserved
0x1D0	LANE_SEL_A0[2:0]	Selects OLDI Port A Lane 0 output pins	0b000: Output from A0 0b001: Output from A1 0b010: Output from A2 0b011: Output from A3 0b100: Output from ACLK 0b101: Reserved 0b110: Reserved 0b111: Reserved
0x1D1	LANE_SEL_B1[2:0]	Selects OLDI Port B Lane 1 output pins	0b000: Output from B0 0b001: Output from B1 0b010: Output from B2 0b011: Output from B3 0b100: Output from BCLK 0b101: Reserved 0b110: Reserved 0b111: Reserved
0x1D1	LANE_SEL_A1[2:0]	Selects OLDI Port A Lane 1 output pins	0b000: Output from A0 0b001: Output from A1 0b010: Output from A2 0b011: Output from A3 0b100: Output from ACLK 0b101: Reserved 0b110: Reserved 0b111: Reserved

ADI Confidential for Newvision Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

PRBS registers are presented in Table 187; refer to the PRBS Testing section for additional information.

Table 187. PRBS Registers

Register Address	Bitfield	Description	Decode
0x1CD	VPRBS_FAIL	Video PRSB pass/fail.	0b0: Video PRBS check passed 0b1: Video PRBS check failed
0x1CD	VPRBS_CHK_EN	Enables video PRBS checker.	0b0: Video PRBS checker disabled 0b1: Video PRBS checker enabled

Table 188 contains registers used to enable the LUT; reference the Color Lookup Table (LUT) section for additional information.

Table 188. LUT Registers

Register Address	Bitfield	Description	Decode
0x1CD	LUT_C_EN	Enables Color C lookup table (LUT) [23:16]	0b0: Color C LUT disabled 0b1: Color C LUT enabled
0x1CD	LUT_B_EN	Enables Color B lookup table (LUT) [15:8]	0b0: Color B LUT disabled 0b1: Color B LUT enabled
0x1CD	LUT_EN_A	Enables Color A lookup table (LUT) [7:0]	0b0: Color A LUT disabled 0b1: Color A LUT enabled

Sync signals can be viewed on GPIO01 if enabled with the registers in Table 189. Reference device-specific datasheet for pin mapping details.

Table 189. GPIO Sync Signal Output

Register Address	Bitfield	Description	Decode
0x1CD	HS_OUT_EN	Output HSYNC from GPIO.	0b0: HSYNC not output 0b1: Output HSYNC from GPIO
0x1CF	VS_OUT_EN	Output VSYNC from GPIO.	0b0: VSYNC not output 0b1: Output VSYNC from GPIO

ADI Confidential for New vision under NDA
Analog Devices Confidential Distributed only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

41.4.1 Programming Examples

41.4.1.1 Stream at Single OLDI Output

Perform the following writes in the OLDI deserializer:

1. OLDI_OUTSEL = Select LVDS output port (A or B) to put the LVDS video data
2. OLDI_FORMAT = Select the mapping format (OLDI or VESA) for the LVDS port
3. OLDI_4TH_LANE = Select 24- or 18-bit output (i.e., 4- or 3-lane output port)
4. OLDI_DUP = Select if the same video should be duplicated on both LVDS ports

41.4.1.2 Stream at Dual OLDI Output

Perform the following writes in the OLDI deserializer:

1. OLDI_FORMAT = Select the mapping format (OLDI or VESA) for the LVDS port
2. OLDI_4TH_LANE = Select 24- or 18-bit output (i.e., 4- or 3-lane output port)
3. OLDI_SPL_MODE = 3
4. OLDI_SPL_EN = 1
5. OLDI_SWAP_AB = Swap routing of even or odd pixels to ports A and B
6. OLDI_SPL_POL = 0

41.4.1.3 Swap Port Lane Mappings and Invert Signal

Swap port A lanes 1 and 2, and invert lane 2 polarity. Perform the following writes in the OLDI deserializer:

1. LANE_SEL_A1 = 2 (lane 1 pins output internal lane 2)
2. LANE_SEL_A2 = 1 (lane 2 pins output internal lane 1)
3. LANE_INV_A2 = 1 (invert polarity of lane 2 pins)

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

41.5 Status and Debug Registers

41.5.1 Status and Error Bits

The following register bits provide status and error information related to LVDS output.

VID_LOCK – (register VIDEO_RX8) – Indicates that a valid video stream is received for the selected channel ID, the video pipeline is locked, and that the video stream is properly recovered by the LVDS to drive the display.

VID_PKT_DET – (register VIDEO_RX8) – Indicates that a stream of video packet for the selected stream ID is detected by the deserializer.

DUAL_OLDI_AUTO_RST_ALIGNED – (register VRX46) – This bitfield reports if the OLDI output has reset itself after detecting a non-alignment on the OLDI outputs. Only applicable if DUAL_OLDI_AUTO_RST_ALIGN = 1.

OUT_STUCK_ERR_FLAG – (register INTR3) – Flag is asserted when the OLDI output is stuck either low or high. Enable with the OUT_STUCK_ERR_OEN bit. The flag is cleared when read if the output stuck event has been resolved.

OUT_OPEN_ERR_FLAG – (register INTR3) – Flag is asserted when the OLDI output is open. Enable with the OUT_OPEN_ERR_OEN bit. The flag is cleared when read if the output open even has been resolved.

Refer to the GMSL2 Error Reporting (ERRB Pin) section for additional information.

The following video receiver status registers and interrupts may also be used:

- Line CRC interrupt (LCRC_ERR_FLAG).
- Video block length error interrupt (VID_BLK_LEN_ERR).
- Video sequence number error (VID_SEQ_ERR).

41.5.2 Debug Procedure

If the OLDI deserializer is not performing as expected, perform the following sequence:

1. Ensure that minimum and maximum frequency limits have not been exceeded.
2. Verify that a valid video stream is detected by the deserializer (VID_PKT_DET).
3. Verify that a valid video stream is received and recovered for the selected channel ID and that it is properly recovered by the LVDS (VID_LOCK).
4. Ensure that the OLDI split control signal is properly mapped to the falling edge of the DE signal (OLDI_SPL_MODE and OLDI_SPL_POL).
5. Ensure that OLDI ports A and B are properly mapped (OLDI_SWAP_AB).
6. Set DUAL_OLDI_AUTO_RST_ALIGN = 1 if pixels are being swapped between OLDI Tx ports in Dual OLDI mode applications.
7. Check that the 4th LVDS lane is not disabled for 24-bit video (OLDI_4TH_LANE).

Maxim Confidential
Provided to XYZ Company Confidential
Advance Information Distribute Only to Customers Under NDA
MAX Confidential for Newvision under NDA

41.6 Feature Availability by Device Family

Dual OLDI is available only on GMSL2 OLDI deserializers (Table 190).

Table 190. Dual OLDI Deserializer Device Family Part Numbers

Device Family	Part Numbers
OLDI Deserializers	MAX96752 MAX96754

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42 eDP/DP Deserializers

42.1 Overview

GMSL2 eDP/DP deserializers convert serial link input to embedded DisplayPort (eDP) or DisplayPort (DP) output. This section provides configuration instructions for the full family of GMSL2 eDP/DP deserializers as eDP/DP source devices with a focus on the DisplayPort interface. All registers used to program the video and DP interface parameters are identical across the device family (see *Feature Availability by Device*).

This family of devices also supports other specialized features, such as DSC, FEC, and HDCP2.3. Dedicated sections elsewhere in this document discuss the configuration and use of these functions.

In the *Operation* section, important system design considerations for implementing GMSL2 eDP/DP deserializers as eDP/DP source devices and ensuring interoperability with eDP/DP sink devices are discussed.

The *Configuration* section provides instructions for use of the eDP/DP controller for successful link training to an eDP or DisplayPort display. This section also details eDP/DP link training and its implementation in GMSL2 eDP/DP deserializers.

The *Debug* section provides instructions to troubleshoot in cases where video is not working.

42.2 Operation

GMSL2 eDP/DP deserializers convert single- or dual-link GMSL2 input to eDP/DP. All eDP/DP deserializers are designed to operate as eDP/DP source devices (as defined by the VESA DisplayPort 1.4 specification).

Note: eDP/DP deserializers are compatible with DisplayPort versions 1.2 through 1.4.

42.2.1 DisplayPort 1.4 Feature Support

GMSL2 eDP/DP deserializers support a subset of eDP and DP features, including single stream transport (SST) mode, hot plug detect (HPD), AUX channel, and spread-spectrum clocking (SSC). Configurable DP link parameters support 1, 2, or 4 active lanes over the range of 1.62 to 8.1Gbps. The GMSL2 eDP/DP deserializers also support pre-emphasis settings of 0, +2, +4, and +6dB and output swing levels (signal amplitudes) of 200 to 450mV (set in 50mV steps), both of which are set automatically during DisplayPort link training, to adapt to different DisplayPort channels (see *Video Timing Regenerator (VTRG)*

The eDP/DP deserializer video timing regenerator (VTRG) is designed to reduce the horizontal video timing variation introduced when video is transferred across the GMSL2 link. The VTRG block is positioned in the eDP/DP deserializer video pipeline immediately after the Video Rx block, where GMSL2 packets are converted to a normal parallel video stream consisting of pixels and sync signals (HS, VS, and DE) (Figure 174).

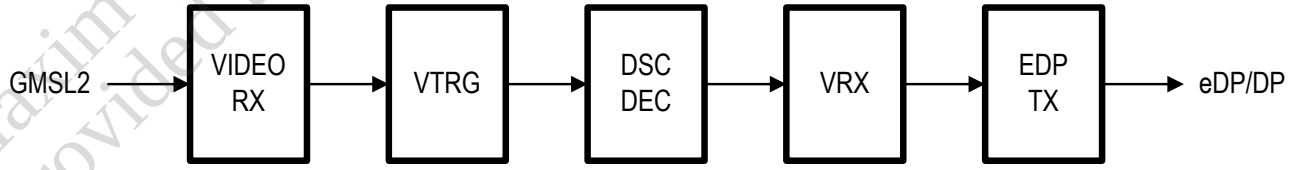


Figure 174. Video Path in GMSL2 eDP/DP Deserializers

Maxim Confidential
Provided to GMSL2 Customers under NDA
Maxim Confidential, Distribution Limited to GMSL2 Customers under NDA

42.2.1.1 VTRG Support by Device Revision

It is crucial to verify the device revision when using the VTRG function. The original generation of production ICs does not support some of the functionality described here.

42.2.1.1.1 Revision 4

Devices that are dev_rev = 4 or less do not support the updated VTRG functionality. For dev_rev = 4, only legacy mode VTRG is supported. For these devices, refer to the Legacy VTRG Configuration sub-section at the end of this section. All information included prior to the Legacy VTRG Configuration sub-section is not relevant to revision 4 and older devices.

Note: Limitations of revision 4 devices

No auto-VTRG support
No Pseudo PCLK support -> Legacy only VTRG
No VPG support
For DSC applications, the remainder of H_{active}/3 cannot equal 2

42.2.1.1.2 Revision 5

Devices that are dev_rev = 5 and newer support pseudo pclk VTRG operation. Other than legacy applications, pseudo PCLK operation should always be used. VPG is supported in this revision. All legacy VTRG configuration is supported for backward compatibility.

Note: Limitations of revision 5 devices

No auto-VTRG support

42.2.1.1.3 Revision 6

Devices that are dev_rev = 6 and newer support both pseudo PCLK VTRG operation and auto-VTRG enable. Other than legacy applications, pseudo PCLK and auto-VTRG should always be used. VPG is supported in this revision. All legacy VTRG configuration is supported for backward compatibility.

Note: Limitations of revision 6 devices

None

42.2.1.2 Overview

The VTRG is a feature that is unique to the eDP deserializer family. It's purpose is to regulate the flow of video data through the internal video path, resulting in uniform data transmission to downstream sink devices. This feature is uniquely required for this device family because the entire video path from the time the video is serialized in the opposite side of the link to the to the time it is transmitted at the display port output consists of a series of fixed bandwidth signal paths that typically provide significantly higher throughput than is required by most video streams. The resulting video flow may result in unsteady transmission of data in which there are long periods of mostly idle video transmission.

As video bandwidth is reduced relative to the available throughput, the length of these idle periods and the resulting timing uncertainty becomes larger, making the data flow regulation of the VTRG function especially important for low bandwidth video applications. The uneven data flow present when the VTRG is not used can

Maxim Confidential
Provided to GMSL2_Customers Under NDA
Maxim Integrated Company Confidential, Do Not Distribute
ADI Confidential for New Vision under NDA

require unusually large amounts of video data to be buffered in downstream devices, potentially disrupting video flow and the resulting displayed image. When the VTRG is enabled, the unsteady data flow can be smoothed considerably, resulting in minimal timing uncertainty and a nice, steady flow of video to the DP sink device.

42.2.1.3 Pseudo PCLK VTRG vs Legacy VTRG Operation

The preferred operating mode for the VTRG function is referred to as pseudo PCLK mode. This function is supported only on device revisions 5 and newer. Older devices only support legacy VTRG mode. Legacy mode is supported by all devices, but it should only be used in conjunction with newer devices if there is a requirement to maintain legacy software. The pseudo PCLK mode is much simpler to manually configure, and it has significantly better data flow rate control. In addition, an automatic configuration function is provided to automatically enable the function as part of DP link training. The legacy VTRG operation and configuration are described in detail at the end of this section in a dedicated subsection. All other information contained within this section is specific to the pseudo PCLK VTRG operating mode.

42.2.1.4 VTRG Automatic Configuration via eDP Controller

Recommended for device revision 6 and newer

The VTRG can automatically be enabled as part of the DP link training process. In this case, the DP controller automatically sets up and enables the VTRG based on the video timing parameters specified as part of the DP link training. The additional register writes detailed below can simply be considered as supplementary information for the link training process. The steps required to utilize the auto-VTRG configuration vary slightly depending on whether the system configuration utilizes DSC. All applications using revision 6 and newer devices should utilize the automatic VTRG mode.

42.2.1.4.1 Auto VTRG Configuration Details

There are two steps to program the VTRG when DSC is not being used:

1. Set the pixel clock frequency in KHz.
2. Enable automatic VTRG operation. Must be performed just prior to issuing the link training command.

There are three steps to program the VTRG when DSC is being used:

1. Set the pixel clock frequency in KHz.
2. Calculate and set the DSC H_{active} value. DSC H_{active} must always be specified when DSC is enabled regardless of whether H_{active} is divisible by 3.
3. Enable automatic VTRG operation. Must be performed just prior to issuing the link training command with DSC already enabled.

Table 192 details VTRG configuration registers that are used during auto configuration

Table 192. VTRG Configuration Registers

Register Address	Bitfield	Value	Register Description
0x7014 0x7015 0x7016	pix_rate_per[7:0] pix_rate_per[15:8] pix_rate_per[23:16]	Program with the pixel rate clock frequency in KHz.	Pixel rate clock frequency

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution only to Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

0xE7C4 0xE7C5	HACT_DSC_VTRG[7:0] HACT_DSC_VTRG[15:8]		This value is required when using the auto VTRG functionality along with DSC. To calculate this value, take the actual horizontal resolution and divide by three. If there is a remainder, round up to the next integer. Take this value and multiply by three to get HACK_DSC_VTRG[15:0].
0xE7C6	AUTO_VTRG_EN[0]	0x01	Video Timing Regenerator (VTRG) enable

42.2.1.4.1.1 Example VTRG Scripts

Two scripts are provided for VESA standard 1080p with and without DSC compression enabled. The register configuration contained within the below examples should be implemented as part of the DP link training. All register writes should occur prior to initiation of DP link training.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.2.1.4.1.1.1 Example Script 1 – 1080p without DSC

```
# Example VTRG Script for no DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 148.50
# Htotal = 2200
# Hactive = 1920
# Hfp = 88
# Hsw = 44
# Hbp = 148
# Vtotal = 1125
# Vactive = 1080
# Vfp = 4
# Vsw = 5
# Vbp = 36
#
# Pixel Clock Rate in KHz - 148,500 (0x024414)
0x90,0x7014,0x14
0x90,0x7015,0x44
0x90,0x7016,0x02
#
# HACT_DSC_VTRG
# only required for DSC applications using auto VTRG
# 0xE7C4 – configure if DSC is used
# 0xE7C5 – configure if DSC is used
#
# AUTO VTRG Enable
0x90,0xE7C6,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
#
# DP link training
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.2.1.4.1.1.2 Example Script 2 –1888x1728 with DSC

```
# Example VTRG Script for using DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 218.726MHz
# Htotal = 2048
# Hactive = 1888
# Hfp = 48
# Hsw = 32
# Hbp = 80
# Vtotal = 1780
# Vactive = 1728
# Vfp = 4
# Vsw = 10
# Vbp = 38
#
# Pixel Clock Rate in KHz - 218,726 (0x035666)
0x90,0x7014,0x66
0x90,0x7015,0x56
0x90,0x7016,0x03
# Calculate and set HACT_DSC_VTRG[15:0]
# HACT_DSC_VTRG = (Hactive/3) rounded up, if necessary, to nearest integer then multiplied by
3
# 1888/3 = 629.33 -> rounded up to 630 then 630*3 = 1890 = 0x0762
0x90,0xE7C4,0x62
0x90,0xE7C5,0x07
# AUTO VTRG Enable
0x90,0xE7C6,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
#
# DP link training
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

42.2.1.5 VTRG Function Details

The VTRG block and its location in the video path are detailed in Figure 175 and Figure 176. In principle, the VTRG acts as a FIFO with a controlled output rate that lies between asynchronous clock domains of the received GMSL2 video and the downstream video path. The VTRG's primary purpose is regulating the video transfer such that it is uniform in the downstream video path. To this end, a rate control function and control loop are used to release a continuous stream of data at the pixel buffer output that effectively has a stable PCLK that matches the average rate of the incoming video.

A video timing generator driven by the rate control subsystem is utilized to generate accurate video timing information based on the recovered PCLK. The timing of the incoming video is used in the control of the pixel buffer, but it does not propagate through the pixel buffer, hence the VTG is employed to insert correct timing information into the outgoing video stream. Video in the pixel buffer is synchronized via the incoming video timing signals such that the first pixel in the buffer at the beginning of transmission is always the first pixel of a frame. Based on this convention of synchronization, the VTG then inserts timing signals into the outgoing video that are consistent with the programmed MSA parameters and are precisely aligned with the outgoing video stream.

In addition to synchronizing the VTG, the control loop monitors the pixel buffer fill status to observe the actual PCLK of the incoming video and adjusts the rate control PCLK accordingly to maintain constant output data rate. In addition to driving data output from the pixel buffer, the rate control PCLK also drives the VTG and the downstream DSC decoder rate control function.

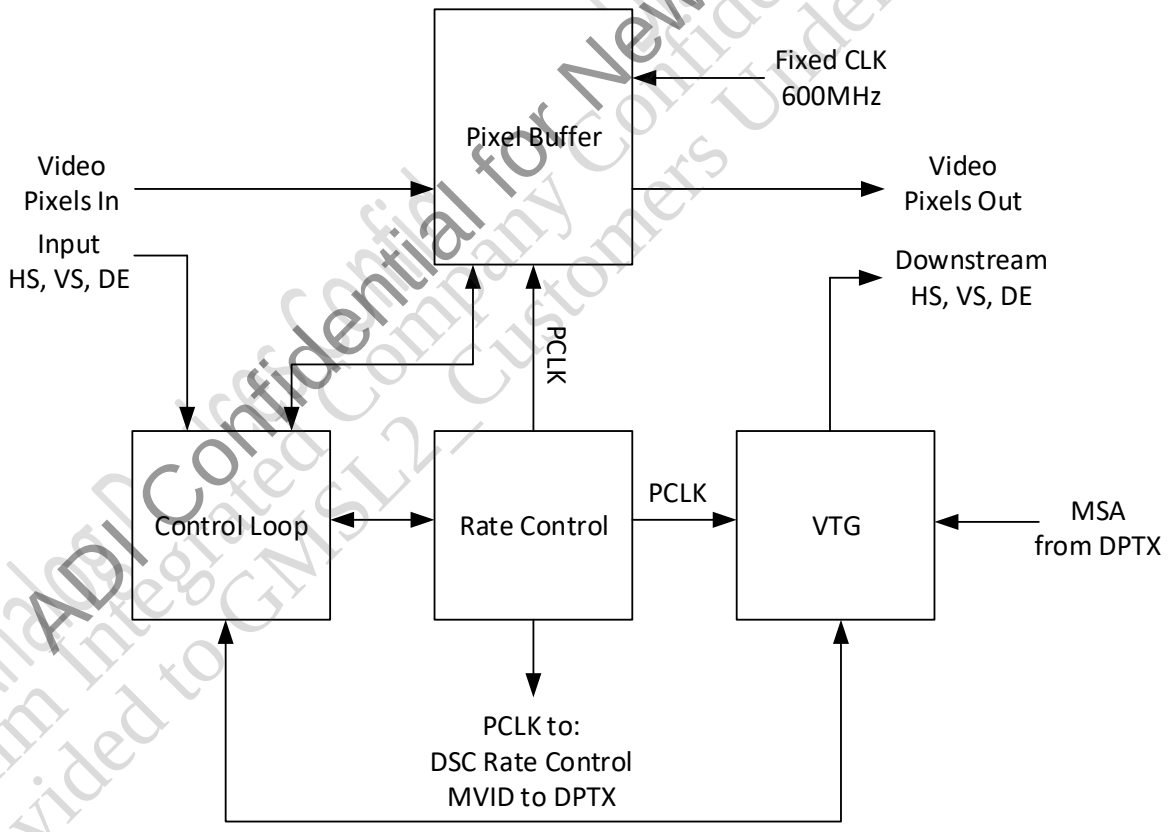


Figure 175. VTRG Block Diagram

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

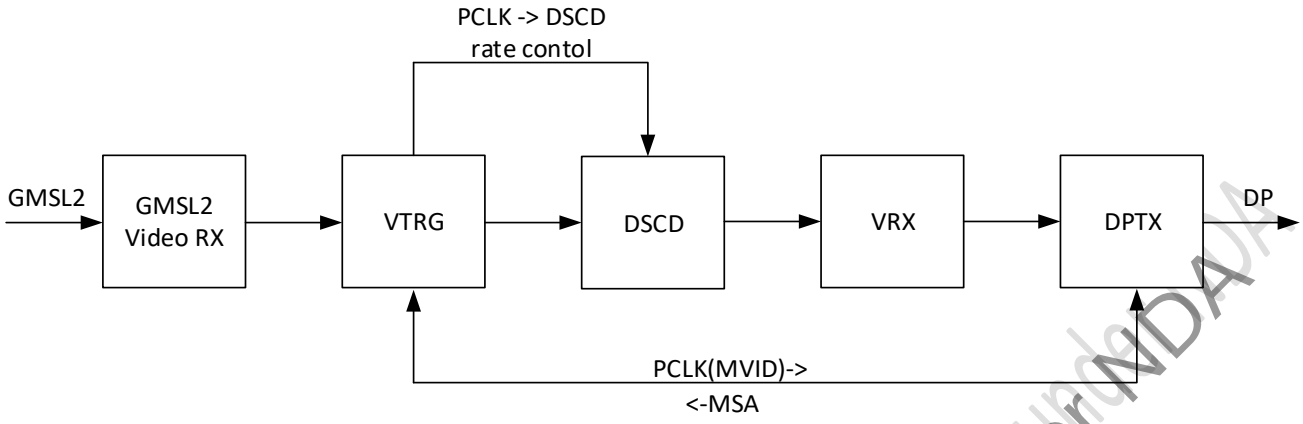


Figure 176. eDP/DP Deserializer Video Path Including PCLK Rate Control

42.2.1.5.1 DPTX Dynamic MVID Tracking

An optional dynamic MVID tracking function is available. When enabled, the VTRG rate control PCLK frequency is forwarded to the DPTX, which then evaluates the nominal MVID depending on the DP configuration, enabling precise real time adjustment of the MVID that is sent to the DP sink. As the rate control PCLK is adjusted to track variation in the incoming video rate, the MVID is automatically updated in the DPTX, ensuring MVID/PCLK vs DP lane rate error never exceeds a few hundred ppm regardless of PCLK drift. This function should be enabled in most applications. Refer to Manual Configuration of the VTRG section for details regarding how to enable dynamic MVID tracking.

42.2.1.5.2 MSA Auto Load from DPTX

In most cases, the nominal video timing parameters used by the DPTX are also used by the VTG. The exception to this is applications that utilize DSC with HACTIVE/3 ≠ integer. This situation is discussed in detail later in this section. In all other cases, the MSA configuration is typically automatically loaded into the VTRG following DP link training. This minimizes redundant configuration, simplifying programming and reducing the likelihood of a mismatch. This function should be enabled in most applications. Refer to Manual Configuration of the VTRG section for details regarding how to enable MSA auto-loading.

42.2.1.6 Manual Configuration of the VTRG (not using eDP Controller Auto Config)

A minimal set of registers must be programmed to initialize the VTRG function. The preferred configuration of the VTRG is to enable both MSA auto loading and DPTX dynamic MVID tracking. All manual configuration discussed here can be handled automatically by the DPTX controller. If eDP controller configuration is employed, none of the register configuration detailed here is required. Refer to the VTRG Configuration via eDP Controller section for additional information. Manual VTRG configuration should only be employed in special cases where the automatic configuration mechanism cannot be used.

42.2.1.6.1 Typical Manual Configuration Sequence

1. Program all required registers for application (including DP, excluding only VTRG)
2. Initiate DP link training (MSA parameters will be loaded into DPTX; required for MSA auto-load)
3. Enable DSC rate control (**if DSC is enabled**) (0x4507=0x80)
4. Configure nominal video PCLK (0x7016:0x7014=PCLK in kHz)
5. Specify DSC PCLK divide ratio (**if DSC is enabled**) (0x7012=0x02)
6. Specify VTRG operating mode (0x7013=0xF)

Analog Confidential Information only
Provided to GMSL Customers Under NDA
Maxim Integrated Confidential for New Vision under MDA

7. Adjust VTG HACT **if DSC is enabled** (in auto-VTRG mode, HACT must always be specified with DSC)
    a. Disable MSA auto-load (after auto-loading MSAs in step 6 above) (0x7013 = 0x0B)
    b. Overwrite VTG HACT with (HACT/3 + (HACT%3 > 0)) * 3 (0x7047:0x7046)
8. Enable VTRG (0x7000 = 0x01)

Note: In the above process that steps 3, 5, and 7 are only considered when DSC is enabled.

An example configuration script with DSC enabled and HACT % 3 ≠ 0 is shown below.

```
# program device and initiate link training as
# a predecessor to this configuration
#
# enable pixel rate control
0x90,0x4507,0x80
# nominal PCLK in kHz
# fill in xx below
0x90,0x7014,0xxx
0x90,0x7015,0xxx
0x90,0x7016,0xxx
#pclk / 3 for DSC
0x90,0x7012,0x02
# enable VTRG with pseudo PCLK, PCLK tracking, auto-MSA load, and dynamic DPTX MVID
0x90,0x7013,0x0F
# hact/3 remainder != 0
# disable DPTX to VTRG auto MSA load
0x90,0x7013,0x0B
# update VTG HACT with roundup(HACT / 3) * 3
0x90,0x7046,0xxx
0x90,0x7047,0xxx
# enable VTRG
0x90,0x7000,0x01
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.2.1.6.2 Configuration Register Details

Table 193. VTRG Configuration Registers

Register	Register/Bitfield	Description
0x4507[7]	PIX_RATE_EXT_EN	DSC Decoder external pixel rate control (only programmed when DSC is used) 0 – Native DSC rate control engine is used for DSC rate control 1 – VTRG rate control engine is used for DSC rate control (recommended)
0x7014[7:0]	PIX_RATE_PER_B0	Nominal PCLK in kHz (this should match actual PCLK of source) Example: if PCLK is 148.5MHz = 148500kHz -> 0x024414 PIX_RATE_PER_B0 = low byte PIX_RATE_PER_B1 = middle byte PIX_RATE_PER_B2 = high byte
0x7015[7:0]	PIX_RATE_PER_B1	
0x7016[7:0]	PIX_RATE_PER_B2	
0x7012[1:0]	PIX_RATE_DIV	PCLK divide ratio 0 – divide by 1 (full PCLK, no DSC) 2 – divide by 3 (PCLK/3, DSC) Other settings are not supported (only programmed when DSC is active; default okay w/o DSC)
0x7013[3:0]	PIX_RATE_CLKP_EN 0x7013[0]	VTRG Mode 0 – Legacy VTRG (not recommended) 1 – Pseudo PCLK VTRG mode (recommended)
	PIX_RATE_CLKP_ADJ 0x7013[1]	VTRG Tracking Enable 0 – PCLK tracking disabled, static PCLK = PIX_RATE_PER[B2:B0] 1 – PCLK tracking enabled, rate control will track actual PCLK (recommended)
	VTG_MSA_AUTO_SET 0x7013[2]	VTG MSA auto-load from DPTX 0 – VTG MSAs are not loaded from DPTX 1 – VTG MSAs are automatically loaded from DPTX (recommended; DP link must be trained prior to VTG configuration)
	DPTX_MCNT_OVRIDE 0x7013[3]	DPTX Dynamic MVID Tracking 0 – DPTX does not use MVID derived from VTRG rate control. DP MVID is static and may have large error 1 – DPTX MVID is dynamically updated based on the actual video PCLK detected by the VTRG (recommended)
0x7046[7:0]	VTG_HACT_LO	VTG HACTIVE manual override (only required when DSC is active and HACT/3 ≠ integer) Write 3 * (roundup(HACT/3)) if required VTG_HACT_LO = low byte VTG_HACT_HI = high byte
0x7047[7:0]	VTG_HACT_HI	
0x7000[0]	VTRG_EN	VTRG enable 0 – VTRG disabled 1 – VTRG enabled (recommended)

ADI Confidential for Review under NDA
Maxim Integrated Confidential Distribution
Provided to GMSL2 Customers Under NDA

42.2.1.6.3 Configuring VTRG when DSC is Active

When DSC is used, the DSC PCLK divide ratio must be specified explicitly as /3. This is because the VTRG is located upstream of the DSC decoder, and as a result the PCLK observed by the VTRG is the PCLK of the compressed video. When DSC is disabled, the default divide ratio of 1 can be used. When using DSC, a fixed value of 0x02 must be written to register 0x7012 (0x02 -> PCLK / 3). Register 0x7012 should only use a value of 0x00 (default, non-DSC) or 0x02 (DSC). Other values are not supported.

In addition, if the HACT period of the video is not integer divisible by 3, then the auto-MSA loading must be disabled (only after an initial MSA load completes to configure all other VTG parameters) and the manual HACT override in registers 0x7047:0x7046 must be programmed with next higher HACT that is integer divisible by 3. For example, if the nominal horizontal resolution is 1600, then HACT is manually programmed to 1602, which is integer divisible by 3.

If auto-VTRG enable is used with DSC, HACT must always be specified regardless of whether HACT is integer divisible by 3. In DSC applications that do not utilize auto-VTRG enable, HACT must only be manually specified when HACT is not integer divisible by 3.

Finally, when DSC is used, the DSC rate control function should be clocked from the VTRG rate control PCLK generator. This ensures the most uniform possible data flow downstream.

All associated registers are detailed in the Configuration Register Details Section

42.2.1.6.4 Additional Configuration Registers (if not using MSA auto-loading)

All VTG parameters can be manually programmed. In most cases, the only VTG parameter that is manually programmed is HACT, and that is only required if DSC is used and HACT/3 is non-integer. All other timing parameters can be automatically loaded via the MSA auto-load from DPTX. It is possible to manually write all of the VTG timing parameters if desired. These can be specified in registers 0x7044 to 0x7053. Refer to the eDP deserializer data sheet for additional details of these registers. In most applications, these registers are not programmed by the user because they typically must match the DP MSA configuration for proper video transmission.

42.2.1.6.5 Sequencing of VTRG Initialization

The VTRG configuration should occur after the DP link is trained. This is required to utilize the MSA auto-load function to configure the VTG from the DPTX (video timing is loaded into the DPTX as part of the DP link training process).

42.2.1.7 VTRG Status Indicators and Debug

The VTRG includes status indicators that can flag disruptions to the data flow.

42.2.1.7.1 Debug Registers

Table 194. VTRG Debug Registers

Register	Register/bit name	Description
0x7008[1:0]	[0] – DE_act_ist [1] – DE_inact_ist	DE activity indicators DE_act_ist is set on each leading edge of DE DE_inact_ist is set on each trailing edge of DE

Analog Devices Confidential, Distribution Only Under NDA
Provided to GMSL2 Customers Under NDA
Maxim Integrated Company Confidential,
ADI Confidential for Review under NDA

		During normal operation, both bits will be continuously set. If they are not set, then the incoming DE may be disrupted
0x7008[3:2]	[2] – VS_act_ist [3] – VS_inact_ist	VS activity indicators VS_act_ist is set on each leading edge of VS VS_inact_ist is set on each trailing edge of VS During normal operation, both bits will be continuously set. If they are not set, then the incoming VS may be disrupted
0x7008[4]	pb_undfl_ist	Pixel Buffer Underflow Indicator 0 – no underflow event (normal operation) 1 – underflow event occurred (problem) An underflow occurs when there is insufficient data in the pixel buffer to satisfy the data flow specified by the downstream rate control PCLK. An underflow condition can occur if there is a sustained difference between the incoming video PCLK and the PCLK generated by the rate control such that the rate control PCLK is higher than the incoming video PCLK. An underflow condition will trigger a reset of the subsystem. In this case, the pixel buffer will begin refilling at the start of the next video frame, and video output will restart when the pixel buffer reaches the nominal fill level.
0x7008[5]	pb_ovfl_ist	Pixel Buffer Overflow Indicator 0 – no overflow event (normal operation) 1 – overflow event occurred (problem) An overflow occurs when there is insufficient space in the pixel buffer to accumulate data arriving from the GMSL2 link video receiver. This can occur if the rate control system pclk output is not correctly tracking the input rate and is outputting data too slowly. Transient changes to the video timing may also be problematic because the rate control loop response is intended to track slow changes, such as PCLK drift. It is not intended to track fast, large transients in the timing. An overflow condition will trigger a reset of the subsystem. In this case, the pixel buffer will begin refilling at the start of the next video frame, and video output will restart when the pixel buffer reaches the nominal fill level.
0x7008[6]	rate_adj_ist	Rate adjust event indicator 0 – No rate adjust event has occurred 1 – Rate adjust event has occurred This bit will typically be set in normal operation when PCLK tracking is enabled, indicating that the control loop is adjusting the rate control PCLK to track the input PCLK. This bit is set every time that the control loop makes a rate adjustment.

ADI Confidential for NEVISION under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customer_Portal, Company Confidential, Under NDA

0x7008[7]	rate_adj_max_ist	Maximum Rate Adjust Event Indicator 0 – No max rate adjust detected (normal operation) 1 – Control loop integral path has reached rate adjust limit If this bit is set, it indicates that the control loop has reached the adjustment limit of the integral control path. Note that the system can still maintain PLK tracking with possibly degraded timing variation based on the control of the proportional control path only. As a result, for modest errors of a couple tenths of a percent beyond the integral control path control range, the system will continue operating more or less normally, although this is not recommended. A max rate adjust condition is typically caused by excessive PCLK error relative to the nominal PCLK.
0x7038[7:0]	PB_CLKP_ADJ_STATUS_B0	Adjusted (actual) PCLK readback in kHz PB_CLKP_ADJ_STATUS_B0 = low byte PB_CLKP_ADJ_STATUS_B1 = middle byte PB_CLKP_ADJ_STATUS_B2 = high bits
0x7039[7:0]	PB_CLKP_ADJ_STATUS_B1	
0x703A[3:0]	PB_CLKP_ADJ_STATUS_B2	

42.2.1.7.2 Interpreting the Status Indicators

If bits 7, 5, or 4 of register 0x7008 are set, then there is a likely problem with the VTRG configuration or the video PCLK. Verify that the actual video PCLK matches the expected value. The VTRG function requires actual PCLK to vary no more than approximately 0.25% from the expected value. Bits 6 and 3:0 of register 0x7008 are typically set during normal operation. If bits 3:0 are not set, it is possible that that a valid video stream is not being received by the GMSL2 input or the VS/DE of this stream are masked in the serializer. All status indicators in register 0x7008 are "write-to-clear." To refresh a status bit, overwrite it with a value of 1. To refresh the entire register, write a value of 0xFF.

42.2.1.7.3 Adjusted PCLK Reporting

The adjusted (actual) PCLK of the video is reported by the VTRG in registers 0x703A:0x7038. This can be a useful debug tool to readily identify PCLK error. In addition, if PCLK adjustments are required elsewhere in the system, the reported value can be used as a convenient means of measuring the absolute PCLK. This value should very precisely represent the actual video PCLK with the primary uncertainty being the crystal frequency reference used by the deserializer.

42.2.1.7.4 Frequency Tolerance

Set the nominal PCLK (PIX_RATE_PER_B[2:0]) to match actual video stream as closely as possible to minimize the PCLK error. The error should be no more than +/- 0.25%. If there is systematic error in the source PCLK, adjust the specified PCLK to match the actual PCLK as closely as possible. As long as the 0.25% tolerance is not violated, the VTRG should lock to the incoming video, and if dynamic DPTX MVID is enabled, the VTRG will then provide a corrected MVID to the sink via the DPTX, ensuring the MVID used by the sink always closely matches the actual PCLK.

Maxim Confidential for New Vision under NDA
Provided to GMSL2 Customers Under NDA

42.2.1.7.5 Measuring HS and VS Uncertainty

One way to observe the impact of the VTRG is to probe the HS and VS signals of the deserializer. They can be routed to GPIO pins for this purpose. With the VTRG disabled, the variation of the period of both HS and VS can be observed with an oscilloscope. With the VTRG enabled, these same measurements can be repeated. This experiment can be conducted for any video timing. Using DSC typically results in the worst timing uncertainty, so DSC should be enabled as a way to create worst case HS and VS uncertainty for a given video timing.
The following table summarizes the typical HS and VS variation observed with and without the VTRG. Note that without the VTRG, uncertainty is typically on the order of 100 pixels while with the VTRG enabled, timing uncertainty is on the order of 1 to 5 pixels. Note that the pseudo PCLK VTRG suppresses both horizontal and vertical timing uncertainty. In contrast, the legacy VTRG mode suppresses horizontal timing uncertainty at the expense of more vertical timing uncertainty.

Table 195. HS and VS Standard Deviations for Two Common Resolutions

Video Resolution	HS Period Stdev VTRG Enabled	HS Period Stdev VTRG Disabled	VS Period Stdev VTRG Enabled	VS Period Stdev VTRG Disabled
1920x1080P60 PCLK=148.5MHz	1.4ns	184ns	5.4ns	136ns
3840x2160P60 PCLK=594MHz	1ns	81ns	6ns	58ns

42.2.1.8 VTRG Video Pattern Generator (VPG)

The VTRG block includes a video pattern generator that can insert a checkerboard test pattern in place of the video stream received from the serializer. A serializer must be connected with active video transmission to enable DP link training when the VPG is used (refer to the VTRG VPG Limitations section for details). However, the VPG generated video pattern, timing, and PCLK are totally independent of the video received from the serializer. The VPG function is only supported by revision 5 (dev_rev=5) and newer devices.

The timing associated with the VPG generated image is automatically generated by the VTG based on the parameters specified during DP link training. The configured PCLK, which should be consistent with MVID and NVID as programmed during link training, is specified in registers 0x7014, 0x7015, and 0x7016 in the same way as during normal VTRG operation. It is important to ensure that MVID and NVID are consistent with the 0x7014/0x7015/0x7016 specified PCLK to ensure robust video recovery in the DP sink device.
The checkerboard pattern size and color component mapping are specified in dedicated registers, which are detailed in the next section. The horizontal and vertical checkerboard size are user programmable, and each 8b RGB color element can be specified as either 0xFF or 0x00 for each alternating checkerboard element, enabling a variety of color patterns to be realized.

42.2.1.8.1 VTRG VPG Registers

The following table summarizes the unique registers associated with the VPG function. Note that some registers are common with standard VTRG video timing regeneration operation, and those registers are detailed in the Configuration Register Details section. Register 0x7013 is specified below and in the primary VTRG register details section because the typical settings are different when using VPG mode.

Table 196. VTRG VPG Configuration Registers

Register	Register/bit name	Description
0x7013[3:0]	PIX_RATE_CLKP_EN PIX_RATE_CLKP_ADJ	0x5 – VPG Mode w/o Dynamic MVID Update to DPTX 0xD – VPG Mode w/ Dynamic MVID Update to DPTX

ADI Confidential for new revision under NDA
Maxim Integrated Confidential, Provided to MSL Customers Under NDA

	VTG_MSA_AUTO_SET DPTX_MCNT_OVRIDE	**Refer to Configuration Register Details for comprehensive description of these bits. Discussed here for clarity in the context of VPG operation**
0x7040[2:0]	VTG_MAN_EN 0x7040[2]	VTG Manual Mode Enable 0 – Manual Mode Disabled (VTG controlled by VTRG controller, normal VTRG operation) 1 – Manual Mode Enabled (VPG mode)
	VTG_PCLK_MODE 0x7040[1:0]	VTG PCLK Mode 0b00 - Horz total/sync counter uses pixel clock Horz active counter uses pixel clock 0b01 - Horz total/sync counter uses pixel clock Horz active counter uses pixel valid **(default)** 0b10 - Horz total/sync counter uses pixel valid Horz active counter uses pixel clock 0b11 - Horz total/sync counter uses pixel valid Horz active counter uses pixel valid **(VPG mode)**
0x7044 **To** **0x7053**	VTG timing parameters	Not typically set by user because MSA auto-loading is used for VTG configuration. VTG parameters typically must match MSAs for correct operation. Auto-load is enabled in register 0x7013.
0x7060	[0] – PIX_PAT_EN	Pattern Generator Enabled. When enabled, generated pattern replaces normal received video stream 0 – Pattern generator disabled (normal operation) 1 – Pattern generator enabled (VPG operation)
	[6:4] - PIX_PAT_COLOR	Pixel pattern color enable. Each bit enables a particular color component for the initial pixel value. Pixel values invert every pixel block. Bit 4 - Red Bit 5 - Green Bit 6 - Blue
0x7064	PIX_PAT_HBLK_LO	Pixel pattern horizontal block size bits 7-0. Horizontal block size is in pixels. The pixel value is inverted every block. 0 – 1 pixel 1 – 2 pixels …….
0x7065	PIX_PAT_HBLK_HI	Pixel pattern horizontal block size bits 7-0. Horizontal block size is in pixels. The pixel value is inverted every block.
0x7066	PIX_PAT_VBLK_LO	Pixel pattern vertical block size bits 7-0. Vertical block size is in lines. The pixel value is inverted every block. 0 – 1 line 1 – 2 lines …….
0x7067	PIX_PAT_VBLK_HI	Pixel pattern vertical block size bits 7-0. Vertical block size is in lines. The pixel value is inverted every block.

ADI Confidential for NEW VISION under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

42.2.1.8.2 VPG Configuration Example

```
# Prerequisites
##########################################
# enable serializer video
# establish video lock
# configure DP link parameters and train link
##########################################
#
# DP link training here
#
# configure VTRG/VPG
# PCLK (should match DP link training; example is PCLK=148.5MHz for 1080P60)
0x90,0x7014,0x14
0x90,0x7015,0x44
0x90,0x7016,0x02
# configure VTRG(no PCLK tracking, MSA auto-load)
0x90,0x7013,0x0D
# manual VTG enable
0x90,0x7040,0x07
# enable pat gen with RGB black/white (all 111s and all 000s alternating)
0x90,0x7060,0x71
# checkerboard size (112 x 112)
0x90,0x7064,0x6F
0x90,0x7065,0x00
0x90,0x7066,0x6F
0x90,0x7067,0x00
# enable vtrg
0x90,0x7000,0x01
```

42.2.1.8.3 VTRG VPG Limitations

1. The DP controller is programmed to check video lock as a prerequisite to training the DP link and enabling DP video transmission. This requires that an upstream serializer with video transmission enabled is connected to the DP deserializer. The serializer to deserializer video transmission is only required to enable the DP controller to observe video lock, and the specific timing and content of the video received from the serializer has no effect whatsoever on the video pattern or timing that is generated by the VPG.

2. The DP controller will automatically configure some VTRG settings when used. As a result, it is recommended to configure and enable the VPG after link training to ensure all configuration is as desired and nothing has been unexpectedly changed by the DP controller.

3. The timing of the video generated by the VTG must match the MSA configuration of the DP output. In most cases, this is handled automatically by the auto-MSA load function, although the user can manually configure all timing parameters in the VTRG. This is not truly a limitation of the VPG. Rather, it is simply mandated by the requirement that the DP video stream timing must match that of the DP MSA configuration. MSA information must be configured via link training prior MSA information being automatically loaded into VTRG if the automatic load function is utilized.

API Confidential for NewVision under NDA
Analog Confidential - Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL Customers Under NDA

42.2.1.9 Legacy VTRG Configuration (required for rev. 4 and older devices)

Legacy VTRG operation must be used on all devices prior to revision 5. Newer devices do support the legacy operating mode, but in the case of newer devices it should only be used in situations where backward compatibility with legacy software must be maintained.

42.2.1.9.1 Legacy VTRG Limitations

Legacy VTRG operation has some notable limitations when compared to the pseudo PCLK VTRG:

1. Pseudo PCLK VTRG suppresses both horizontal and vertical timing uncertainty while legacy VTRG suppresses horizontal timing uncertainty at the expense of increased vertical timing uncertainty.
2. Horizontal Resolution Limitation with DSC
   Legacy VTRG functionality in device revision 4 (DEV_REV=4) and older have the following limitation. Newer devices utilizing the legacy VTRG function do not have this limitation.

$$roundup\left(\frac{H_{active}}{3}\right) = even$$

42.2.1.9.2 Legacy VTRG Configuration Details

There are four steps to program the VTRG:

3. Setup pixel rate control
4. Setup video timing generator (VTG)
5. Adjust control loop parameters
6. Enable VTRG

Steps 1–3 require registers to be programmed with the exact video timings for the given use case. There is a tool in the GMSL GUI that automatically generates the required configuration for steps 1-4 above. The following table detail all registers that are written.

Table 197. Programming Steps to Enable VTRG

Register Address	Bitfield	Value	Register Description
0x4507	pix_rate_ext_en[0]	Set high to enable	Enables external control. If using DSC, must be set after loading DSC profile
0x7010	pix_rate_clk_act[4:0]	Program with "Pix rate clks/act reg value" from c-code (in hex)	Pixel rate clocks per active cycle
0x7010	pix_rate_ctrl_en[0]	set bit 7 high (don't overwrite pixel rate clks/act)	Pixel rate control enable
0x7011	pix_rate_inact_red[7:0]	Program with "Pix rate red reg value" from c-code (in hex)	Pixel rate inactive-to-inactive redistribution count
0x7012	pix_rate_div[1:0]	Set to 0x00 for no DSC set to 0x02 if using DSC	Pixel rate divide factor to match sparse pixel rates, if DSC compression enabled divide by 3, else divide by 1

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential, Distribute only under NDA
Provided to GMSL2 Customers Under NDA

0x7014 0x7015	pix_rate_act_lo[7:0] pix_rate_act_hi[1:0]	Program with "Pix rate active reg value" from c-code (in hex)	Pixel rate active count bits 7–0
0x7016 0x7017	pix_rate_inact_lo[7:0] pix_rate_inact_hi[2:0]	Program with "Pix rate inactive reg value" from c-code (in hex)	Pixel rate active count bits 7–0
0x7044 0x7045	vtg_htot_lo[7:0] vtg_htot_hi[7:0]	Program with "Horiz total" from c-code (in hex)	VTG horizontal total (line length) bits 7–0, programmed in numbers of 600Mhz clocks
0x7046 0x7047	vtg_hact_lo[7:0] vtg_hact_hi[7:0]	Program with actual Hactive in pixels (divide by 3 is DSC is used)	VTG horizontal active bits 7–0, programmed in numbers of pixels or compressed bytes (pix/3)
0x7048 0x7049	vtg_hstt_lo[7:0] vtg_hstt_hi[7:0]	Program with "Horiz start" from c-code (in hex)	VTG horizontal sync start (sw + bp) bits 7–0, programmed in numbers of 600Mhz clocks
0x704A 0x704B	vtg_hsw_lo[7:0] vtg_hsw_hi[7:0]	program with "Hsync width" from c-code. Set MSB for positive sync polarity	VTG horizontal sync width bits 7–0, programmed in numbers of 600Mhz clocks. MSB is sync polarity.
0x704C 0x704D	vtg_vtot_lo[7:0] vtg_vtot_hi[7:0]	Program with Vtotal in lines	VTG vertical total (frame length) bits 7–0, programmed in numbers of horizontal lines
0x704E 0x704F	vtg_vact_lo[7:0] vtg_vact_hi[7:0]	Program with Vactive in lines	VTG vertical active bits 7–0, programmed in numbers of horizontal lines
0x7050 0x7051	vtg_vstt_lo[7:0] vtg_vstt_hi[7:0]	Program with Vsw + Vbp (sync width plus back porch)	VTG vertical sync start (sw + bp) bits 7–0, programmed in numbers of horizontal lines
0x7052 0x7053	vtg_vsw_lo[7:0] vtg_vsw_hi[7:0]	Program with Vsw. Set MSB for positive sync polarity	VTG vertical sync width bits 7–0, programmed in numbers of horizontal lines
	Additional VTRG settings to optimize performance	Set following registers 0x7020 -> 0x07 0x702D -> 0x01 0x702E -> 0x01 0x702F -> 0x01 0x7030 -> 0x04	Additional VTRG settings to optimize performance
0x7000	vtrg_en[0]	0x01	Video Timing Regenerator (VTRG) enable

ADI Confidential for Newvision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

42.2.1.9.2.1 Example VTRG Scripts

Two Legacy VTRG example scripts are provided for VESA standard 1080p with and without DSC compression enabled.

42.2.1.9.2.1.1 Example Script 1 – 1080p with no DSC

```
# Example VTRG Script for no DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 148.50
# Htotal = 2200
# Hactive = 1920
# Hfp = 88
# Hsw = 44
# Hbp = 148
# Vtotal = 1125
# Vactive = 1080
# Vfp = 4
# Vsw = 5
# Vbp = 36
#
# Enable external pixel rate control
0x90,0x4507,0x80
# Pixel Rate Clks/Act - program to "Pix rate clk/act" from c-code & set MSB to enable
0x90,0x7010,0x83
# Pixel Rate Redist – program with “Pix rate red Reg value” from c-code
0x90,0x7011,0x00
# Pixel Rate Divide - program to 0x00 if no DSC, program to 0x02 if DSC used
0x90,0x7012,0x00
# Pixel Rate Active - program with "Pix rate active Reg value" from c-code
0x90,0x7014,0x62
0x90,0x7015,0x00
# Pixel Rate inactive - program with "Pix rate active Reg value" from c-code
0x90,0x7016,0x2C
0x90,0x7017,0x01
#
# VTG Horizontal Total - program with "Horiz total" from c-code
0x90,0x7044,0x9B
0x90,0x7045,0x22
# VTG Horizontal Active - program with Hactive (divide by 3 is DSC used)
0x90,0x7046,0x80
0x90,0x7047,0x07
# VTG Horizontal Sync Start - program with "Horiz start" from c-code
0x90,0x7048,0x0A
0x90,0x7049,0x03
# VTG Horizontal Sync Width & Polarity - program with "Hsync width" from c-code.
# Set MSB for positive sync polarity
0x90,0x704A,0x58
0x90,0x704B,0x80
# VTG Vertical Total - Program with Vtotal in lines
0x90,0x704C,0x65
0x90,0x704D,0x04
# VTG Vertical Active - Program with Vactive in lines
0x90,0x704E,0x38
0x90,0x704F,0x04
# VTG Vertical Sync Start - Program with Vsw + Vbp
0x90,0x7050,0x29
0x90,0x7051,0x00
# VTG Vertical Sync Width - Program with Vsw
# Set MSB for positive sync polarity
0x90,0x7052,0x05
```

Maxim Integrated Confidential Distribute only under NDA
Provided to GMSL2_Customers Under NDA
Maxim Confidential for New Vision under NDA

```
0x90,0x7053,0x80
#
# Additional VTRG settings (optimizations)
0x90,0x7020,0x07
0x90,0x702D,0x01
0x90,0x702E,0x01
0x90,0x702F,0x01
0x90,0x7030,0x04
#
# VTRG Enable
0x90,0x7000,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.2.1.9.2.1.2 Example Script 2 – 1080p with DSC

```
# Example VTRG Script for using DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 148.50
# Htotal = 2200
# Hactive = 1920
# Hfp = 88
# Hsw = 44
# Hbp = 148
# Vtotal = 1125
# Vactive = 1080
# Vfp = 4
# Vsw = 5
# Vbp = 36
#
# Enable external pixel rate control
0x90,0x4507,0x80
# Pixel Rate Clks/Act - program to "Pix rate clk/act" from c-code & set MSB to enable
0x90,0x7010,0x83
# Pixel Rate Redist – program with “Pix rate red Reg value” from c-code
0x90,0x7011,0x00
# Pixel Rate Divide - program to 0x00 if no DSC, program to 0x02 if DSC used
0x90,0x7012,0x02
# Pixel Rate Active - program with "Pix rate active Reg value" from c-code
0x90,0x7014,0x62
0x90,0x7015,0x00
# Pixel Rate inactive - program with "Pix rate active Reg value" from c-code
0x90,0x7016,0x2C
0x90,0x7017,0x01
#
# VTG Horizontal Total - program with "Horiz total" from c-code
0x90,0x7044,0x9B
0x90,0x7045,0x22
# VTG Horizontal Active - program with Hactive (divide by 3 is DSC used)
0x90,0x7046,0x80
0x90,0x7047,0x02
# VTG Horizontal Sync Start - program with "Horiz start" from c-code
0x90,0x7048,0x0A
0x90,0x7049,0x03
# VTG Horizontal Sync Width & Polarity - program with "Hsync width" from c-code.
# Set MSB for positive sync polarity
0x90,0x704A,0x58
0x90,0x704B,0x80
# VTG Vertical Total - Program with Vtotal in lines
0x90,0x704C,0x65
0x90,0x704D,0x04
# VTG Vertical Active - Program with Vactive in lines
0x90,0x704E,0x38
0x90,0x704F,0x04
# VTG Vertical Sync Start - Program with Vsw + Vbp
0x90,0x7050,0x29
0x90,0x7051,0x00
# VTG Vertical Sync Width - Program with Vsw
# Set MSB for positive sync polarity
0x90,0x7052,0x05
0x90,0x7053,0x80
#
# Additional VTRG settings (optimizations)
0x90,0x7020,0x07
0x90,0x702D,0x01
```

Maxim Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
Maxim Confidential - Distribution only under NDA
Confidential for Newvision under NDA

```
0x90,0x702E,0x01
0x90,0x702F,0x01
0x90,0x7030,0x04
#
# VTRG Enable
0x90,0x7000,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
```

42.2.1.9.3 Legacy VTRG Debug

The primary indicator that the VTRG is operating correctly is by checking the pixel buffer underflow/overflow interrupt statuses, pb_undfl_ist and pb_ovfl_ist (register 0x7008) (.). Since these are latching bits and may have been inadvertently set at startup, first clear the bits by writing them to 1. Then, read the bits to check for underflow/overflow interrupts. Underflow or overflow may occur if:

- VTG configuration does not match the original video mode timings.
  - Verify that the VTG settings match the original video mode timings. In particular, check the calculated values for horizontal total, horizontal start, and horizontal sync width.
- Pixel rate control configuration does not match the original video nominal pixel clock.
  - Use the calculator tool to verify settings.
  - Check that pix_rate_div is correct for DSC/non-DSC video.
- Pixel buffer control loop adjustments are too tight for the original video mode (i.e., pb_rate_adj_max is set too low for the allowed original video pixel clock PPM offset from nominal).
  - Check the rate_adj_max_ist interrupt status. This is a useful indicator that adjustments have reached the maximum limit. It is not abnormal for it to be set, but if it is not set that indicates that there is additional margin available on the rate adjustment.
  - Repeatedly read the live rate adjustment status, vtg_ht_adj. This live bit indicates if the rate adjustment is centered or skewed to the negative or positive side.
  - Read the rate adjustment min/max watermark, vtg_ht_adj_min/max. These provide information regarding the minimum and maximum excursions for rate adjustment, especially for the very wide default setting for pb_rate_adj_max. Wide excursions may indicate that the gain settings (pb_rate_adj_step_* settings) are too high and causing instability in the control loop. It is also an indication of the jitter on the incoming video.

Table 198. Legacy VTRG Debug Registers

Register Address	Bitfield	Description
0x7000	vtrg_en[0]	Enables VTRG operation
0x7008	pb_undfl_ist[0]	This latching bit indicates that the pixel buffer underflowed. It must be initially cleared (write 1 to clear) after VTRG enable. It will remain cleared when the VTRG is operating properly.
0x7008	pb_ovfl_ist[0]	This latching bit indicates that the pixel buffer overflowed. It must be initially cleared (write 1 to clear) after VTRG enable. It will remain cleared when the VTRG is operating properly.
0x7008	rate_adj_ist[0]	This latching bit indicates that a rate adjust event occurred. It must be initially cleared (write 1 to clear) after VTRG enable. It is normal to be set when the VTRG is operating properly.

ADI Confidential Information Now Revision under NDA
Maxim Integrated Confidential Information NDA
Provided to GMSL2_ Customer under NDA

0x7008	rate_adj_max_ist[0]	This latching bit indicates that a rate adjust event was limited by the maximum setting (pb_rate_adj_max). It must be initially cleared (write 1 to clear) after VTRG enable. It is not abnormal to be set when the VTRG is operating properly but does warn that the adjustment limit is being reached.
0x7010	pix_rate_ctrl_en[0]	Enables pixel rate control that provides a pixel data valid to control the pixel flow rate associated with the video pipeline clock. The pixel data valid paces pixel output at the nominal pixel clock rate.
0x7020	pb_del_acc_per[3:0]	Sets the period (number of horizontal lines) for accumulating the pixel buffer fill delta information used to make rate adjustments. This value is the zero-relative number of lines. This is the control loop bandwidth.
0x702C	pb_rate_adj_step_0[5:0]	Sets the rate adjust step size for range 0. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 0.
0x702D	pb_rate_adj_step_1[5:0]	Sets the rate adjust step size for range 1. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 1.
0x702E	pb_rate_adj_step_2[5:0]	Sets the rate adjust step size for range 2. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 2.
0x702F	pb_rate_adj_step_3[5:0]	Sets the rate adjust step size for range 3. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 3.
0x7030	pb_rate_adj_max[6:0]	Sets the maximum allowed rate adjustment. This 2's complement positive value is used as a +/- limit. A value of zero disables the limit. This limits the ppm offset of the pixel clock with respect to the video pipeline clock that can be tracked. It also limits the magnitude of jitter transfer of the input video to output video.
0x7035	vtg_ht_adj[7:0]	VTG horizontal total adjust live status. This is a 2's-complement value.
0x7036	vtg_ht_adj_min[7:0]	VTG horizontal total adjust minimum watermark. This captures and holds the minimum (most negative) horizontal total adjust since the last reset. Writing (any value) will reset the value to zero. This is a 2's-complement value.
0x7037	vtg_ht_adj_max[7:0]	VTG horizontal total adjust maximum watermark. This captures and holds the maximum (most positive) horizontal total adjust since the last reset. Writing (any value) will reset the value to zero. This is a 2's-complement value.

ADI Confidential for Newvision under NDA
Analog Devices Inc. Maxim Integrated Confidential Provided to GMSL2_Customer_Under_NDA

).

DisplayPort features that are not supported by GMSL2 deserializers:

- Multi-Stream Transport (MST) mode.
- DP sink EDID read.
- Full DisplayPort output swing levels.
- Fast AUX link training and no-AUX training support.
- I2C-over-Aux
- DisplayPort HDCP, DSC, and FEC support (Note: these are supported on the GMSL link).

eDP features that are not supported by GMSL2 deserializers:

- Multi-Segment Operation (MSO).
- Advanced Link Power Management (ALPM).
- Panel Self-Refresh (PSR).
- Fast AUX link training and no-AUX training support.

42.2.2 eDP/DP Interoperability Considerations

For optimum system design, the following parameters should be carefully reviewed and considered to ensure interoperability of GMSL2 eDP/DP deserializers (eDP/DP source device) with eDP/DP sink devices. Due to the effects of serialization and jitter inherent to serial link systems, it is important to consider:

- Video source clock drift,
- DP sink pixel buffer depth
- DP sink clock regeneration method.

The block diagram below (Figure 173) depicts a typical use of the eDP/DP deserializer within a serial link system. Packetized compressed or uncompressed video is transmitted over the GMSL link. Video PCLK is not directly transferred from serializer to deserializer. The eDP/DP deserializer evaluates the frequency of the equivalent PCLK relative to its fixed 600MHz video path and then calculates dynamically changing equivalent MVID (VTRG required), which is transferred to the sink as part of the MSA information associated with the DP video stream.

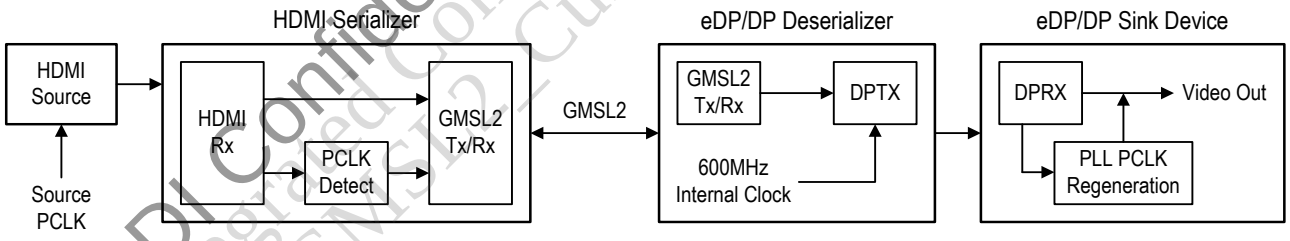


Figure 173. eDP/DP Block Diagram

42.2.2.1 Mvid and Nvid Reporting

The DP clock mode is defined by the relationship between the DPTX clock and the DisplayPort link reference clock. In synchronous mode, the DPTX and DisplayPort link share the same reference clock. In asynchronous mode, the DPTX clock is independent of the DisplayPort link reference clock. The GMSL2 eDP/DP deserializers operate in DP asynchronous clock mode.

The Mvid and Nvid values are 24-bit values that describe the pixel clock (PCLK) relative to the DP link rate:

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

$$\frac{Mvid}{Nvid} = \frac{PCLK}{100 * DP\ Link\ Rate}$$

The Mvid and Nvid values are used to regenerate the PCLK in the eDP/DP sink device. According to the VESA DisplayPort 1.4 specification:

> In Asynchronous Clock mode, the Sink device shall use the received Mvid and Nvid values as a hint for regenerating the pixel clock. The final stream clock regeneration shall be realized so that the regenerated video active line and frame timings are in sync with blanking end (BE) framing control link symbol reception… How a Sink device uses (or even does not use) Mvid and Nvid values for stream clock regeneration is implementation-specific.

In asynchronous clock mode, Nvid is specified to be a static value and Mvid is set by the user according to the source PCLK. According to the DP standard, the Mvid value is a hint for regenerating the pixel clock. The specification requirement for accuracy is that Mvid/Nvid ratio for a given link rate should match the pixel rate with ±1% accuracy.

GMSL2 eDP/DP deserializers transfer MVID and NVID in real time to the sink as part of the MSA information. Per specification, a static Nvid value is reported.

MVID can either be specified as a static value that does not track actual PCLK or it can be dynamically evaluated and updated to match the actual received PCLK frequency. When dynamic PCLK tracking is enabled (enabled by default with VTRG), the MVID will be updated in real time to match actual PCLK with an uncertainty that is on the order of the crystal reference uncertainty. If dynamic MVID tracking is disabled (VTRG off), it is crucial to ensure that the source PCLK does not drift by more than 1% from the PCLK implied by the hard coded MVID that is specified when link training parameters are configured. Some sinks may be less tolerant of PCLK to MVID error, and as a result it is strongly recommended to enable dynamic MVID tracking to ensure the error never exceeds a few hundred ppm.

42.2.2.2 Blanking Variation

There is some variation in the video blanking timings at the DP sink interface (including horizontal blanking variation and vertical sync variation) inherent to the GMSL2 link. Video timing variation is caused by:

- Video packetization and de-packetization over the serial link. This causes a horizontal blanking variation of ±72 pixels.
- Packet scheduling across the serial link. The delay introduced by packet transmit scheduling can be variable and is dependent on the total bandwidth usage across the link and the GMSL2 link rate. Other data transmitted on the GMSL2 link increases the traffic into the GMSL2 packet scheduler, which results in increased variability of the transmission of video packets across the link. A higher PCLK value will also result in increased pixel variability: for the highest rated PCLK of 600MHz, the variation caused is up to ±70 pixels.

The result of these two effects is a total GMSL2 blanking variability of up to ±142 pixels. This variation is further increased with additional downstream variations associated with DisplayPort packetization in the deserializer, DP lane skew/deskew, and depacketization in the DP sink device. To accommodate the combined variability, it is

Maxim Confidential
Provided to GMSL2_Customers under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA
Any GMSL Confidential Info to Contribute to Upcoming Revision under NDA

recommended to use a pixel buffer at least 512 pixels deep (±256 pixels) in the DP sink device to regenerate the original video timing.

Enabling the VTRG function significantly suppresses timing uncertainty such that it typically becomes negligible.

42.2.2.3 Video Timing Regenerator

GMSL2 eDP/DP deserializers include a Video Timing Regenerator (VTRG) function to significantly reduce timing uncertainty. The VTRG suppresses horizontal variation in typical cases by >90% while simultaneously also slightly reducing vertical timing uncertainty. All applications using revision 6 and newer devices should utilize the automatic VTRG mode.

Note: The VTRG is a different feature than the VTG (Video Timing Generator) in GMSL2 serializers. See the VTRG section for more information.

42.2.2.4 Horizontal Resolution Limitation

GMSL2 eDP/DP deserializers do not support odd horizontal resolutions (i.e., H_{active} must be an even number of pixels). In addition, the following condition must be met if DSC and the legacy VTRG are used in combination as may be the case with older device revisions:

$$roundup\left(\frac{H_{active}}{3}\right) = even$$

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

42.3 Configuration

The eDP/DP deserializer must be configured after each power-up as follows.

1. Configure all general parameters of the serializer and deserializer (control channels, DSC, GPIOs, etc.).
2. Set video configuration registers and DP link parameters.
3. Enable source video and confirm video flow in GMSL devices
4. Write command to deserializer to initiate link training (full link training command).

Note: A GUI tool is available to aid in eDP/DP configuration. Reference the GUI Document for more information.

42.3.1 eDP/DP Link Training Overview

Link training is a procedure defined in the DisplayPort specification that occurs between any eDP/DP source and sink during initialization, after the source device detects a sink device driving the HPD pin high. Link training consists of two phases: a clock recovery phase and a channel equalization/symbol-lock/inter-lane alignment phase.

For additional information on eDP/DP, refer to the official VESA DisplayPort Specification (verison 1.4).

42.3.1.1 GMSL2 eDP/DP Link Training Implementation

An internal programmable state machine executes the link training procedure for all GMSL2 eDP/DP deserializers. The state machine executes instructions stored in the deserializer's One Time Programmable (OTP) memory. These instructions are factory-installed and are not user-accessible.

eDP/DP link training is triggered when the following three conditions are met:

- The "Start Link training command" has been sent to the deserializer
- The HPD pin is high
- Video lock is high

Note: Video must be active prior to initiating link training.

42.3.1.2 eDP/DP Controller Operation

The eDP/DP controller is user readable and writable via the main control channel port (i.e., I^2C or UART), either locally on the deserializer or remotely from the serializer.

The command-based eDP/DP controller allows several operations with the eDP/DP sink device over the eDP/DP AUX channel. Commands are selected in register 0xE776 and executed by writing 0x80 to register 0xE777. Available commands include:

- Reboot
- Start Full Link Training
- Read AUX Channel
- Write AUX Channel

Note: In typical operation, only the "Start Full Link Training" command needs to be used.

ADI Confidential for New Revision under NDA
Maxim Integrated Company Confidential, Provided to GMSL Customers Under NDA

42.3.1.2.1 Reboot

This command forces a reboot of the eDP/DP controller. To issue reboot command, write the following registers:

- Command select: 0x90,0xE776,0x01
- Execute command: 0x90,0xE777,0x80

42.3.1.2.2 Start Full Link Training

This is the most common command for the eDP/DP controller; in typical use, it is the only command that is needed. This command runs full link training on the eDP/DP sink device using the video and DP parameters. Configure all DP link configuration registers, including spread spectrum and ASSR, prior to issuing the link training command. This command initializes the AUX channel, eDP/DP Tx PHYs, and eDP/DP link training. To issue command, write the following registers:

- Command select: 0x90,0xE776,0x02
- Execute command: 0x90,0xE777,0x80

42.3.1.2.3 Read AUX channel

This command reads a DisplayPort Configuration Data (DPCD) register on the eDP/DP sink device. The register address is specified by the user in addresses 0xE778 and 0xE779. Once the command has executed, the return data (a byte) is stored in 0xE77A. The AUX channel must be configured prior to using the command (this occurs at power-up). As an example, to read DPCD sink register 0x0100 (main link bandwidth setting), write the following registers:

- LSBs of read address: 0x90,0xE778,0x00
- MSBs of read address: 0x90,0xE779,0x01
- Command select: 0x90,0xE776,0x10
- Execute command: 0x90,0xE777,0x80
- LSBs of return value: read 0x90,0xE77A

42.3.1.2.4 Write AUX channel

This command writes a DPCD register on the eDP/DP sink device. The register address is specified by the user in address 0xE778 and 0xE779. The data (a byte) to be written is specified in 0xE77A. The AUX channel must be configured prior to using the command (this occurs at power-up). The example below writes DPCD sink register 0x0100 with data 0x0A. To issue command, write the following registers:

- LSBs of write address: 0x90,0xE778,0x00
- MSBs of write address: 0x90,0xE779,0x01
- LSBs of data to write: 0x90,0xE77A,0x0A
- Command select: 0x90,0xE776,0x20
- Execute command: 0x90,0xE777,0x80

Analog Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to CVAST Customers Under NDA
Advance Confidential for New Version under NDA

42.3.2 DisplayPort Configuration Details

The video configuration registers for video timing and eDP/DP output settings are primarily located in register addresses 0xE790 to 0xE7DF. A limited number of additional configuration bits are located elsewhere in the register map (see *Table 191* below). These registers should be programmed with the desired DP link parameters (DP link rate, number of lanes, etc.) and the video resolution for the given application.

Note: A script to program these registers can either be created manually or automatically using the GUI script generator. Reference the GUI Document for more information.

42.3.2.1 DP Bandwidth and Lane Count/Rate Configuration

Ensure that the DP link bandwidth available is greater than the bandwidth required for the video resolution.

$$\frac{(DP_Link_Rate \ * \ DP_Lane_Count)}{encoding_overhead} \geq \text{PCLK} * \text{bpp}$$

Where the DisplayPort encoding overhead is 1.25 (due to 8b10b encoding) and bits per pixel (bpp) is equal to 24 for RGB888 video data.

42.3.2.2 Video Timing Parameters

All video timing parameters must be entered in the appropriate registers prior to link training. These parameters are sent to the DP sink as part of the video stream MSA information, and they are also copied to the VTRG automatically as part of the link training process. The timing parameters should exactly match the anticipated timing of the video source in most cases.

42.3.2.3 Output Spread Spectrum

The DisplayPort standard supports 0.5% down spread as a means of mitigating emissions issues. The eDP/DP family of deserializers supports this function with spread spectrum integrated into the DP link training process of the DP controller.

When enabling spread spectrum, the spread level must be adjusted in register 0x6003[2:0] (config_spread_bit_ratio) to specify 0.25% center spread (0.5% down spread) prior to issuing the link training command. Following correct configuration of config_spread_bit_ratio, spread spectrum can be enabled as part of link training by setting the SSC_ENABLE bit in register 0xE7B1 prior to issuing the link training command.

42.3.2.4 CMU Lock Check Setting

Prior to link training, the DP controller must be configured to wait for the DP link CMU to lock. This is accomplished by setting SS_ENABLE = 1 in register 0xE7B0. This bit must be set to 1 in all cases regardless of DP port configuration.

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribution Only Under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

42.3.2.5 Lane Alignment Check

The DP lane alignment check is disabled by default in the DP controller. In most cases, this check should be enabled prior to the link training command being issued. To enabled lane alignment check, set NO_ALIGN_CHK=0 in register 0xE7DE.

42.3.2.6 VTRG

The VTRG is used to smooth data flow and reduce timing uncertainty that may be created by the fixed speed, packetized nature of data transmission in the GMSL link. This function is typically enabled automatically during link training as configured by registers 0x7014, 0x7015, 0x7016, 0xE7C4, 0xE7C5, and 0xE7C6.

42.3.2.7 Alternate Scrambler Seed Reset (ASSR)

The Alternate Scrambler Seed Reset (ASSR) capability for embedded DisplayPort (eDP) provides a means of display authentication for content protection on inaccessible eDP links (i.e. no external cabling). This is simply a way to differentiate between inaccessible (embedded) links vs accessible links that could expose content that requires protection.

Normal DisplayPort operation scrambles data that is passed across the main link, primarily to reduce EMI. To properly descramble the data, the DPTX and DPRX scramblers are synchronized by the DPTX periodically sending scrambler reset symbols. This reinitializes the scrambler and descrambler by setting each to the reset seed value (0xFFFF). When ASSR is enabled, the alternate reset seed is used (0xFFFE). ASSR is an eDP feature that is intended to be enabled only on inaccessible DP links, thus, differentiating accessible links from inaccessible.

The ASSR feature is enabled or disabled in both the DPTX and DPRX. The eDP deserializer includes the DPTX function. The DPRX is found in the sink device. The sink device must support ASSR as it must be simultaneously enabled in both devices for proper operation.

42.3.2.7.1 ASSR Configuration Process

Configure ASSR prior to link training.

Step 1 (check sink/DPRX support for ASSR):
DPTX (eDP serializer) must interrogate DPRX (sink) and verify ASSR support by checking DPRX DPCD 0x000D[0]. If this bit has a value of 1, then sink DPRX supports ASSR. If this bit is 0, then DPRX does not support ASSR and it should not be enabled.

Step 2 (enable ASSR in sink/DPRX):
To enable ASSR in DPRX, set DPCD 0x010A[0] =1.
ASSR can also be disabled if required by setting DPCD 0x010A[0] = 0.

Step 3 (enable ASSR in source/DPTX/eDP SER):
ALTERNATE_SCRAMBLER_RESET (0x611C[0]) = 1 to enable DPTX ASSR
ALTERNATE_SCRAMBLER_RESET (0x611C[0]) = 0 to disable DPTX ASSR

Step 4:
Train DP Link

ADI Confidential for Nevision under NDA
Analog Devices Confidential Information Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

42.3.2.8 DisplayPort Configuration Registers

Table 191. eDP/DP Deserializer Register Table

Register Address	Bitfield	Register Description	Decode
0x6003	config_spread_bit_ratio	DP PLL spread spectrum triangle wave generator deviation Only a setting of 1 (0.5% down spread) is DP compliant. All other settings are not recommended	0x0: SS disabled 0x1: 0.25% center / 0.5% down spread 0x2: 0.5% center / 1% down spread 0x3: 1% center / 2% down spread 0x4: 2% center / 4% down spread 0x5: 4% center / 8% down spread 0x6: 4% center / 8% down spread 0x7: 4% center / 8% down spread
0x611C	ALTERNATE_SCRAMBLER_RESET	Enable DPTX ASSR	0x0: ASSR Disabled 0x1: ASSR Enabled
0x7014	PIX_RATE_PER_B0	Nominal video pclk (kHz) low byte (VTRG)	Nominal video PCLK in kHz
0x7015	PIX_RATE_PER_B1	Nominal video pclk (kHz) middle byte (VTRG)	Nominal video PCLK in kHz
0x7016	PIX_RATE_PER_B2	Nominal video pclk (kHz) high byte (VTRG)	Nominal video PCLK in kHz
0xE790	Link_Rate[7:0]	eDP/DP data rate per lane. eDP/DP spec lists four possible link rates: 1.62, 2.7, 5.4, and 8.1Gbps per lane. For enough bandwidth, ensure:	0x0006: 1.62Gbps per lane 0x000A: 2.7Gbps per lane 0x0014: 5.4Gbps per lane 0x001E: 8.1Gbps per lane
0xE791	Link_Rate[15:8]		
0xE792	Lane_Count[7:0]	eDP/DP lane count. Number of active lanes. eDP/DP spec lists three possible values: 1, 2, or 4 lanes.	0x0001: 1 lane 0x0002: 2 lanes 0x0004: 4 lanes
0xE793	Lane_Count[15:8]		
0xE794	Hres[7:0]	Lower 8 bits of the horizontal resolution of the mainstream video source This value is the number of active pixels per line.	0xXX: Lower 8 bits of the horizontal resolution of the mainstream video source
0xE795	Hres[15:8]	Upper 8 bits of the horizontal resolution of the mainstream video source This value is the number of active pixels per line.	0xXX: Upper 8 bits of the horizontal resolution of the mainstream video source
0xE796	Hfp[7:0]	Lower 8 bits of the horizontal front porch of the mainstream video source This value is expressed in pixels.	0xXX: Lower 8 bits of the horizontal front porch of the mainstream video source
0xE797	Hfp[15:8]	Upper 8 bits of the horizontal front porch of the mainstream video source This value is expressed in pixels.	0xXX: Upper 8 bits of the horizontal front porch of the mainstream video source
0xE798	Hsw[7:0]	Lower 8 bits of the horizontal sync width of the mainstream video source This value is expressed in pixels.	0xXX: Lower 8 bits of the horizontal sync width of the mainstream video source
0xE799	Hsw[15:8]	Upper 8 bits of the horizontal sync width of the mainstream video source This value is expressed in pixels.	0xXX: Upper 8 bits of the horizontal sync width of the mainstream video source
0xE79A	Hbp[7:0]	Lower 8 bits of the horizontal back porch of the mainstream video source This value is expressed in pixels.	0xXX: Lower 8 bits of the horizontal back porch of the mainstream video source

ADI Confidential for New vision under NDA
Analog Devices Distributed Maxim Integrated Company Confidential Provided to GMSL2 Customers Under NDA

0xE79B	Hbp[15:8]	Upper 8 bits of the horizontal back porch of the mainstream video source This value is expressed in pixels	0xXX: Upper 8 bits of the horizontal back porch of the mainstream video source
0xE79C	Vres[7:0]	Lower 8 bits of the vertical resolution of the mainstream video source This value is the number of active lines per frame.	0xXX: Lower 8 bits of the vertical resolution of the mainstream video source
0xE79D	Vres[15:8]	Upper 8 bits of the vertical resolution of the mainstream video source This value is the number of active lines per frame.	0xXX: Upper 8 bits of the vertical resolution of the mainstream video source
0xE79E	Vfp[7:0]	Lower 8 bits of the vertical front porch of the mainstream video source This value is expressed in lines.	0xXX: Lower 8 bits of the vertical front porch of the mainstream video source
0xE79F	Vfp[15:8]	Upper 8 bits of the vertical front porch of the mainstream video source This value is expressed in lines.	0xXX: Upper 8 bits of the vertical front porch of the mainstream video source
0xE7A0	Vsw[7:0]	Lower 8 bits of the vertical sync width of the mainstream video source This value is expressed in lines.	0xXX: Lower 8 bits of the vertical sync width of the mainstream video source
0xE7A1	Vsw[15:8]	Upper 8 bits of the vertical sync width of the mainstream video source This value is expressed in lines.	0xXX: Upper 8 bits of the vertical sync width of the mainstream video source
0xE7A2	Vbp[7:0]	Lower 8 bits of the vertical back porch of the mainstream video source This value is expressed in lines.	0xXX: Lower 8 bits of the vertical back porch of the mainstream video source
0xE7A3	Vbp[15:8]	Upper 8 bits of the vertical back porch of the mainstream video source This value is expressed in lines.	0xXX: Upper 8 bits of the vertical back porch of the mainstream video source
0xE7A4	Hwords[7:0]	Number of 16-bit words in an active display line minus the lane count	Calculated as: Hwords = 1.5 * Hres - Lane Count
0xE7A5	Hwords[15:8]		
0xE7A6	Mvid[7:0]	Calculated value for DisplayPort asynchronous mode	Calculated as: (Pclk * 32768)/(DP Link Rate * 100) *Where Pclk is in MHz and Link Rate is in Gb/s. See example below.*
0xE7A7	Mvid[15:8]		
0xE7A8	Nvid[7:0]	Is a constant value of 32768	Should be set to 0x8000
0xE7A9	Nvid[15:8]		
0xE7AA	TUC_Value[7:0]	Lower 8 bits of the transfer unit which is a display port packet and represents valid data symbols and stuffing symbols	0xXX: Lower 8 bits of the transfer unit which is a display port packet and represents valid data symbols and stuffing symbols
0xE7AB	TUC_Value[15:8]	Upper 8 bits of the transfer unit which is a display port packet and represents valid data symbols and stuffing symbols Set to 64 (0x0040) for typical use cases.	0xXX: Upper 8 bits of the transfer unit which is a display port packet and represents valid data symbols and stuffing symbols
0xE7AC	HVPOL[7:0]	Horizontal and Vertical Sync Polarity	Bit 0 = Hsync polarity. 0: active high, 1: active low Bit 1 = Vsync polarity. 0: active high, 1: active low
0xE7AD	HVPOL[15:8]		

ADI Confidential For New Version Under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA

0xE7B0[0]	SS Enable	Wait for DP CMU lock prior to link training	**Must be set to 1 in all cases regardless of whether spread spectrum is enabled**
0xE7B1[4]	SSC Enable	DP link output spread spectrum Enable If SS is enabled, config_spread_bit_ratio in register 0x6003 must be set to 1 prior to link training to enable correct 0.5% down spread	0 – SS disabled 1 – SS Enabled
0xE7B2	EDP_DPLL_CTRL0_B0[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x0050 (0x0050 for SSC) 2.7Gbps – set to 0x0050 (0x0050 for SSC) 5.4Gbps – set to 0x0030 (0x0030 for SSC) 8.1Gbps – set to 0x0030 (0x0030 for SSC)
0xE7B3	EDP_DPLL_CTRL0_B1[15:8]		
0xE7B4	EDP_DPLL_CTRL0_B2[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x44CC (0x4235 for SSC) 2.7Gbps – set to 0x4000 (0x45D7 for SSC) 5.4Gbps – set to 0x4000 (0x45D7 for SSC) 8.1Gbps – set to 0x4000 (0x45D7 for SSC)
0xE7B5	EDP_DPLL_CTRL0_B3[15:8]		
0xE7B6	EDP_DPLL_CTRL1_B0[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x3081 (0x3081 for SSC) 2.7Gbps – set to 0x206C (0x206B for SSC) 5.4Gbps – set to 0x206C (0x206B for SSC) 8.1Gbps – set to 0x206C (0x206B for SSC)
0xE7B7	EDP_DPLL_CTRL1_B1[15:8]		
0xE7B8	EDP_DPLL_CTRL1_B2[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x1007 (0x1007 for SSC) 2.7Gbps – set to 0x0007 (0x0007 for SSC) 5.4Gbps – set to 0x0014 (0x0014 for SSC) 8.1Gbps – set to 0x0014 (0x0014 for SSC)
0xE7B9	EDP_DPLL_CTRL1_B3[15:8]		
0xE7BA	EDP_CMU_CTRL0_B0[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x0001 (0x0001 for SSC) 2.7Gbps – set to 0x0001 (0x0001 for SSC) 5.4Gbps – set to 0x002E (0x002E for SSC) 8.1Gbps – set to 0x002E (0x002E for SSC)
0xE7BB	EDP_CMU_CTRL0_B1[15:8]		
0xE7BC	EDP_CMU_CTRL0_B2[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x0000 (0x0000 for SSC) 2.7Gbps – set to 0x0000 (0x0000 for SSC) 5.4Gbps – set to 0x0102 (0x0102 for SSC) 8.1Gbps – set to 0x0000 (0x0000 for SSC)
0xE7BD	EDP_CMU_CTRL0_B3[15:8]		
0xE7BE	EDP_DPLL_CTRL0_B0[7:0]	This register must be set depending on the eDP/DP link rate setting in register 0xE790 and 0xE791	1.62Gbps – set to 0x0052 (0x0052 for SSC) 2.7Gbps – set to 0x0052 (0x0052 for SSC) 5.4Gbps – set to 0x0032 (0x0032 for SSC) 8.1Gbps – set to 0x0032 (0x0032 for SSC)
0xE7BF	EDP_DPLL_CTRL0_B1[15:8]		
0xE7C4	HACT_DSC_VTRG[7:0]	This register contains the adjusted HRES value for use with the automatic VTRG function when operating with DSC.	See the section of this document describing the operation of the automatic VTRG function when operating with DSC.
0xE7C5	HACT_DSC_VTRG[15:8]	This register contains the adjusted HRES value for use with the automatic VTRG function when operating with DSC.	See the section of this document describing the operation of the automatic VTRG function when operating with DSC.

ADI Confidential for New Version Under NDA
Maxim Integrated Company Confidential, Distribution Prohibited
Provided to GMSL2 Customers Under NDA

0xE7C6	AUTO_VTRG_ENABLE[0]	This register enables the automatic configuration and use of the VTRG.	0: Auto VTRG is disabled 1: Auto VTRG is enabled
0xE7D1	HPD_DEBOUNCE[0]	Whenever HPD transitions from low to high, the default is to wait 50ms to debounce. For systems that have HPD hardwired from the DP sink device, this debouncing is unnecessary and will increase link training time by 50ms. This debouncing can be eliminated by setting this bit just prior to sending the link training command.	0: Enable 50ms debounce of HPD 1: Disable 50ms debounce of HPD
0xE7DE	No_Align_Chk[0]	Determines whether DP lane alignment is checked after successful symbol lock is achieved. The default is to skip the lane alignment check. It is recommended that lane alignment check should typically be enabled.	0: Check lane alignment after symbol lock (recommended) 1: Skip lane alignment check after symbol lock (default, not recommended)
0xE7DF	DP_MISC (0xE7DF[2:0])	Misc DP configuration options Bit 0: reserved Bit 1: video lock check skip Bit 2: reserved	Bit 0 – always set to 0 Bit 1 – 0: check video lock prior to DP training 1: do not check video lock prior to DP training Bit 2 – always set to 0

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

42.3.3 Video Timing Regenerator (VTRG)

The eDP/DP deserializer video timing regenerator (VTRG) is designed to reduce the horizontal video timing variation introduced when video is transferred across the GMSL2 link. The VTRG block is positioned in the eDP/DP deserializer video pipeline immediately after the Video Rx block, where GMSL2 packets are converted to a normal parallel video stream consisting of pixels and sync signals (HS, VS, and DE) (Figure 174).

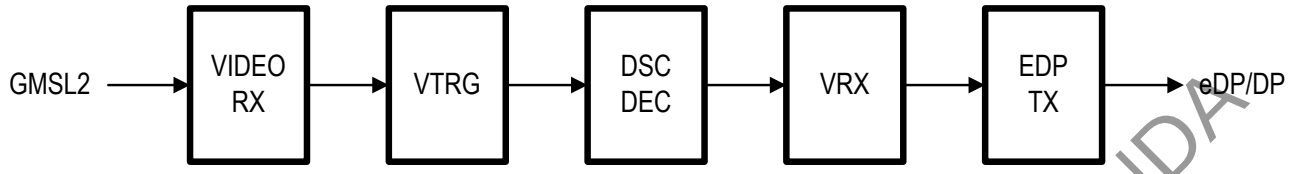


Figure 174. Video Path in GMSL2 eDP/DP Deserializers

42.3.3.1 VTRG Support by Device Revision

It is crucial to verify the device revision when using the VTRG function. The original generation of production ICs does not support some of the functionality described here.

42.3.3.1.1 Revision 4

Devices that are dev_rev = 4 or less do not support the updated VTRG functionality. For dev_rev = 4, only legacy mode VTRG is supported. For these devices, refer to the Legacy VTRG Configuration sub-section at the end of this section. All information included prior to the Legacy VTRG Configuration sub-section is not relevant to revision 4 and older devices.

Note: Limitations of revision 4 devices
No auto-VTRG support
No Pseudo PCLK support -> Legacy only VTRG
No VPG support
For DSC applications, the remainder of H_{active}/3 cannot equal 2

42.3.3.1.2 Revision 5

Devices that are dev_rev = 5 and newer support pseudo pclk VTRG operation. Other than legacy applications, pseudo PCLK operation should always be used. VPG is supported in this revision. All legacy VTRG configuration is supported for backward compatibility.

Note: Limitations of revision 5 devices
No auto-VTRG support

42.3.3.1.3 Revision 6

Devices that are dev_rev = 6 and newer support both pseudo PCLK VTRG operation and auto-VTRG enable. Other than legacy applications, pseudo PCLK and auto-VTRG should always be used. VPG is supported in this revision. All legacy VTRG configuration is supported for backward compatibility.

Note: Limitations of revision 6 devices
None

ADI Confidential for Newvision under NDA
Maxim Proprietary Do not distribute
Provided to GMSL2_Customers Under NDA

42.3.3.2 Overview

The VTRG is a feature that is unique to the eDP deserializer family. It's purpose is to regulate the flow of video data through the internal video path, resulting in uniform data transmission to downstream sink devices. This feature is uniquely required for this device family because the entire video path from the time the video is serialized in the opposite side of the link to the to the time it is transmitted at the display port output consists of a series of fixed bandwidth signal paths that typically provide significantly higher throughput than is required by most video streams. The resulting video flow may result in unsteady transmission of data in which there are long periods of mostly idle video transmission.

As video bandwidth is reduced relative to the available throughput, the length of these idle periods and the resulting timing uncertainty becomes larger, making the data flow regulation of the VTRG function especially important for low bandwidth video applications. The uneven data flow present when the VTRG is not used can require unusually large amounts of video data to be buffered in downstream devices, potentially disrupting video flow and the resulting displayed image. When the VTRG is enabled, the unsteady data flow can be smoothed considerably, resulting in minimal timing uncertainty and a nice, steady flow of video to the DP sink device.

42.3.3.3 Pseudo PCLK VTRG vs Legacy VTRG Operation

The preferred operating mode for the VTRG function is referred to as pseudo PCLK mode. This function is supported only on device revisions 5 and newer. Older devices only support legacy VTRG mode. Legacy mode is supported by all devices, but it should only be used in conjunction with newer devices if there is a requirement to maintain legacy software. The pseudo PCLK mode is much simpler to manually configure, and it has significantly better data flow rate control. In addition, an automatic configuration function is provided to automatically enable the function as part of DP link training. The legacy VTRG operation and configuration are described in detail at the end of this section in a dedicated subsection. All other information contained within this section is specific to the pseudo PCLK VTRG operating mode.

42.3.3.4 VTRG Automatic Configuration via eDP Controller

Recommended for device revision 6 and newer

The VTRG can automatically be enabled as part of the DP link training process. In this case, the DP controller automatically sets up and enables the VTRG based on the video timing parameters specified as part of the DP link training. The additional register writes detailed below can simply be considered as supplementary information for the link training process. The steps required to utilize the auto-VTRG configuration vary slightly depending on whether the system configuration utilizes DSC. All applications using revision 6 and newer devices should utilize the automatic VTRG mode.

42.3.3.4.1 Auto VTRG Configuration Details

There are two steps to program the VTRG when DSC is not being used:

7. Set the pixel clock frequency in KHz.
8. Enable automatic VTRG operation. Must be performed just prior to issuing the link training command.

There are three steps to program the VTRG when DSC is being used:

4. Set the pixel clock frequency in KHz.
5. Calculate and set the DSC H_{active} value. DSC H_{active} must always be specified when DSC is enabled regardless of whether H_{active} is divisible by 3.

ADI Confidential Revision under NDA
Maxim Integrated Confidential, Provided to CMSL Customers under NDA

6. Enable automatic VTRG operation. Must be performed just prior to issuing the link training command with DSC already enabled.

Table 192 details VTRG configuration registers that are used during auto configuration

Table 192. VTRG Configuration Registers

Register Address	Bitfield	Value	Register Description
0x7014 0x7015 0x7016	pix_rate_per[7:0] pix_rate_per[15:8] pix_rate_per[23:16]	Program with the pixel rate clock frequency in KHz.	Pixel rate clock frequency
0xE7C4 0xE7C5	HACT_DSC_VTRG[7:0] HACT_DSC_VTRG[15:8]		This value is required when using the auto VTRG functionality along with DSC. To calculate this value, take the actual horizontal resolution and divide by three. If there is a remainder, round up to the next integer. Take this value and multiply by three to get HACK_DSC_VTRG[15:0].
0xE7C6	AUTO_VTRG_EN[0]	0x01	Video Timing Regenerator (VTRG) enable

42.3.3.4.1.1 Example VTRG Scripts

Two scripts are provided for VESA standard 1080p with and without DSC compression enabled. The register configuration contained within the below examples should be implemented as part of the DP link training. All register writes should occur prior to initiation of DP link training.

ADI Confidential for review under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

42.3.3.4.1.1.1 Example Script 1 – 1080p without DSC

```
# Example VTRG Script for no DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 148.50
# Htotal = 2200
# Hactive = 1920
# Hfp = 88
# Hsw = 44
# Hbp = 148
# Vtotal = 1125
# Vactive = 1080
# Vfp = 4
# Vsw = 5
# Vbp = 36
#
# Pixel Clock Rate in KHz - 148,500 (0x024414)
0x90,0x7014,0x14
0x90,0x7015,0x44
0x90,0x7016,0x02
#
# HACT_DSC_VTRG
# only required for DSC applications using auto VTRG
# 0xE7C4 – configure if DSC is used
# 0xE7C5 – configure if DSC is used
#
# AUTO VTRG Enable
0x90,0xE7C6,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
#
# DP link training
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.3.3.4.1.1.2 Example Script 2 –1888x1728 with DSC

```
# Example VTRG Script for using DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 218.726MHz
# Htotal = 2048
# Hactive = 1888
# Hfp = 48
# Hsw = 32
# Hbp = 80
# Vtotal = 1780
# Vactive = 1728
# Vfp = 4
# Vsw = 10
# Vbp = 38
#
# Pixel Clock Rate in KHz - 218,726 (0x035666)
0x90,0x7014,0x66
0x90,0x7015,0x56
0x90,0x7016,0x03
# Calculate and set HACT_DSC_VTRG[15:0]
# HACT_DSC_VTRG = (Hactive/3) rounded up, if necessary, to nearest integer then multiplied by
3
# 1888/3 = 629.33 -> rounded up to 630 then 630*3 = 1890 = 0x0762
0x90,0xE7C4,0x62
0x90,0xE7C5,0x07
# AUTO VTRG Enable
0x90,0xE7C6,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
#
# DP link training
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

42.3.3.5 VTRG Function Details

The VTRG block and its location in the video path are detailed in Figure 175 and Figure 176. In principle, the VTRG acts as a FIFO with a controlled output rate that lies between asynchronous clock domains of the received GMSL2 video and the downstream video path. The VTRG's primary purpose is regulating the video transfer such that it is uniform in the downstream video path. To this end, a rate control function and control loop are used to release a continuous stream of data at the pixel buffer output that effectively has a stable PCLK that matches the average rate of the incoming video.

A video timing generator driven by the rate control subsystem is utilized to generate accurate video timing information based on the recovered PCLK. The timing of the incoming video is used in the control of the pixel buffer, but it does not propagate through the pixel buffer, hence the VTG is employed to insert correct timing information into the outgoing video stream. Video in the pixel buffer is synchronized via the incoming video timing signals such that the first pixel in the buffer at the beginning of transmission is always the first pixel of a frame. Based on this convention of synchronization, the VTG then inserts timing signals into the outgoing video that are consistent with the programmed MSA parameters and are precisely aligned with the outgoing video stream.

In addition to synchronizing the VTG, the control loop monitors the pixel buffer fill status to observe the actual PCLK of the incoming video and adjusts the rate control PCLK accordingly to maintain constant output data rate. In addition to driving data output from the pixel buffer, the rate control PCLK also drives the VTG and the downstream DSC decoder rate control function.

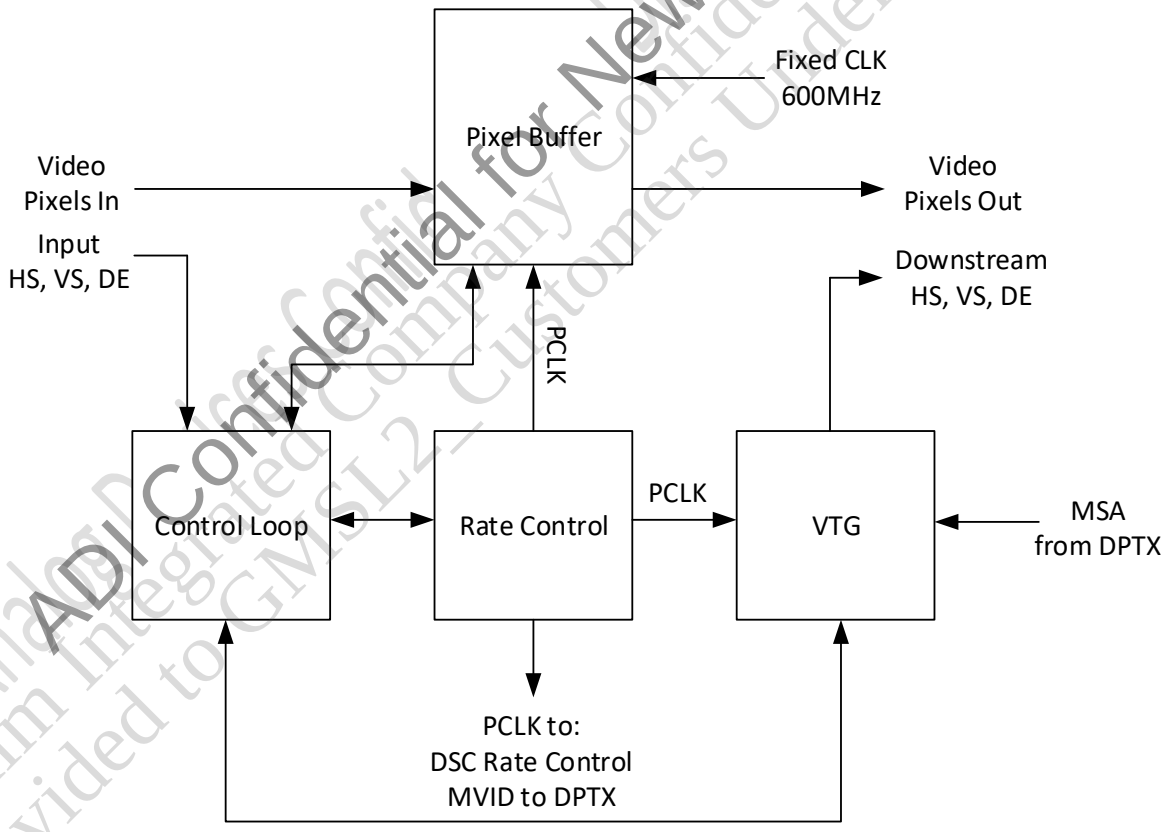


Figure 175. VTRG Block Diagram

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

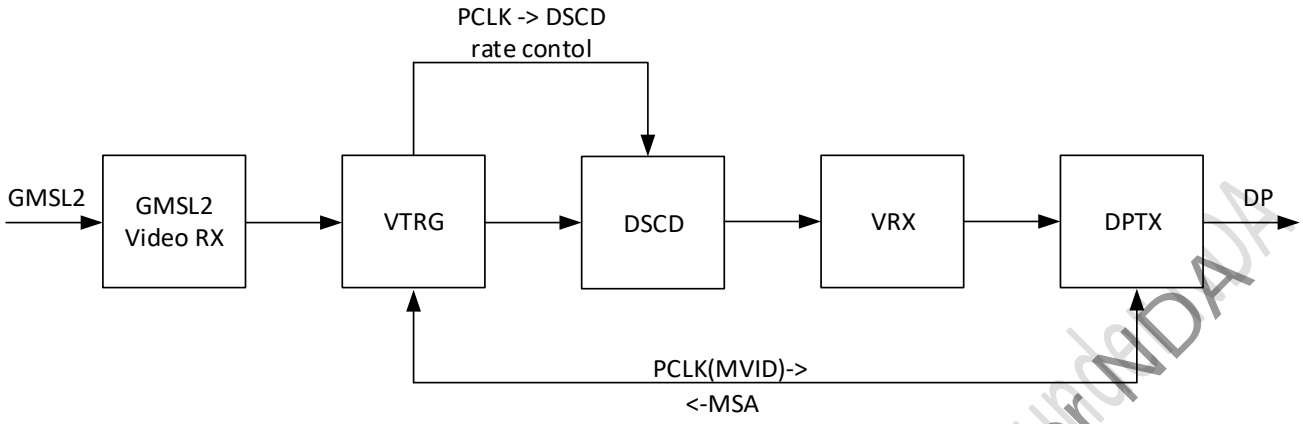


Figure 176. eDP/DP Deserializer Video Path Including PCLK Rate Control

42.3.3.5.1 DPTX Dynamic MVID Tracking

An optional dynamic MVID tracking function is available. When enabled, the VTRG rate control PCLK frequency is forwarded to the DPTX, which then evaluates the nominal MVID depending on the DP configuration, enabling precise real time adjustment of the MVID that is sent to the DP sink. As the rate control PCLK is adjusted to track variation in the incoming video rate, the MVID is automatically updated in the DPTX, ensuring MVID/PCLK vs DP lane rate error never exceeds a few hundred ppm regardless of PCLK drift. This function should be enabled in most applications. Refer to Manual Configuration of the VTRG section for details regarding how to enable dynamic MVID tracking.

42.3.3.5.2 MSA Auto Load from DPTX

In most cases, the nominal video timing parameters used by the DPTX are also used by the VTG. The exception to this is applications that utilize DSC with HACTIVE/3 ≠ integer. This situation is discussed in detail later in this section. In all other cases, the MSA configuration is typically automatically loaded into the VTRG following DP link training. This minimizes redundant configuration, simplifying programming and reducing the likelihood of a mismatch. This function should be enabled in most applications. Refer to Manual Configuration of the VTRG section for details regarding how to enable MSA auto-loading.

42.3.3.6 Manual Configuration of the VTRG (not using eDP Controller Auto Config)

A minimal set of registers must be programmed to initialize the VTRG function. The preferred configuration of the VTRG is to enable both MSA auto loading and DPTX dynamic MVID tracking. All manual configuration discussed here can be handled automatically by the DPTX controller. If eDP controller configuration is employed, none of the register configuration detailed here is required. Refer to the VTRG Configuration via eDP Controller section for additional information. Manual VTRG configuration should only be employed in special cases where the automatic configuration mechanism cannot be used.

42.3.3.6.1 Typical Manual Configuration Sequence

9. Program all required registers for application (including DP, excluding only VTRG)
10. Initiate DP link training (MSA parameters will be loaded into DPTX; required for MSA auto-load)
11. Enable DSC rate control (**if DSC is enabled**) (0x4507=0x80)
12. Configure nominal video PCLK (0x7016:0x7014=PCLK in kHz)
13. Specify DSC PCLK divide ratio (**if DSC is enabled**) (0x7012=0x02)
14. Specify VTRG operating mode (0x7013=0xF)

Analog Confidential for Newvision under MDA
Maxim Integrated Confidential Under NDA
Provided to GMSL Customers Under NDA

15. Adjust VTG HACT **if DSC is enabled** (in auto-VTRG mode, HACT must always be specified with DSC)
    a. Disable MSA auto-load (after auto-loading MSAs in step 6 above) (0x7013 = 0x0B)
    b. Overwrite VTG HACT with (HACT/3 + (HACT%3 > 0)) * 3 (0x7047:0x7046)
16. Enable VTRG (0x7000 = 0x01)

Note: In the above process that steps 3, 5, and 7 are only considered when DSC is enabled.

An example configuration script with DSC enabled and HACT % 3 ≠ 0 is shown below.

```
# program device and initiate link training as
# a predecessor to this configuration
#
# enable pixel rate control
0x90,0x4507,0x80
# nominal PCLK in kHz
# fill in xx below
0x90,0x7014,0xxx
0x90,0x7015,0xxx
0x90,0x7016,0xxx
#pclk / 3 for DSC
0x90,0x7012,0x02
# enable VTRG with pseudo PCLK, PCLK tracking, auto-MSA load, and dynamic DPTX MVID
0x90,0x7013,0x0F
# hact/3 remainder != 0
# disable DPTX to VTRG auto MSA load
0x90,0x7013,0x0B
# update VTG HACT with roundup(HACT / 3) * 3
0x90,0x7046,0xxx
0x90,0x7047,0xxx
# enable VTRG
0x90,0x7000,0x01
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.3.3.6.2 Configuration Register Details

Table 193. VTRG Configuration Registers

Register	Register/Bitfield	Description
0x4507[7]	PIX_RATE_EXT_EN	DSC Decoder external pixel rate control (only programmed when DSC is used) 0 – Native DSC rate control engine is used for DSC rate control 1 – VTRG rate control engine is used for DSC rate control (recommended)
0x7014[7:0]	PIX_RATE_PER_B0	Nominal PCLK in kHz (this should match actual PCLK of source) Example: if PCLK is 148.5MHz = 148500kHz -> 0x024414 PIX_RATE_PER_B0 = low byte PIX_RATE_PER_B1 = middle byte PIX_RATE_PER_B2 = high byte
0x7015[7:0]	PIX_RATE_PER_B1	
0x7016[7:0]	PIX_RATE_PER_B2	
0x7012[1:0]	PIX_RATE_DIV	PCLK divide ratio 0 – divide by 1 (full PCLK, no DSC) 2 – divide by 3 (PCLK/3, DSC) Other settings are not supported (only programmed when DSC is active; default okay w/o DSC)
0x7013[3:0]	PIX_RATE_CLKP_EN 0x7013[0]	VTRG Mode 0 – Legacy VTRG (not recommended) 1 – Pseudo PCLK VTRG mode (recommended)
	PIX_RATE_CLKP_ADJ 0x7013[1]	VTRG Tracking Enable 0 – PCLK tracking disabled, static PCLK = PIX_RATE_PER[B2:B0] 1 – PCLK tracking enabled, rate control will track actual PCLK (recommended)
	VTG_MSA_AUTO_SET 0x7013[2]	VTG MSA auto-load from DPTX 0 – VTG MSAs are not loaded from DPTX 1 – VTG MSAs are automatically loaded from DPTX (recommended; DP link must be trained prior to VTG configuration)
	DPTX_MCNT_OVRIDE 0x7013[3]	DPTX Dynamic MVID Tracking 0 – DPTX does not use MVID derived from VTRG rate control. DP MVID is static and may have large error 1 – DPTX MVID is dynamically updated based on the actual video PCLK detected by the VTRG (recommended)
0x7046[7:0]	VTG_HACT_LO	VTG HACTIVE manual override (only required when DSC is active and HACT/3 ≠ integer) Write 3 * (roundup(HACT/3)) if required VTG_HACT_LO = low byte VTG_HACT_HI = high byte
0x7047[7:0]	VTG_HACT_HI	
0x7000[0]	VTRG_EN	VTRG enable 0 – VTRG disabled 1 – VTRG enabled (recommended)

ADI Confidential for review under NDA
Maxim Integrated Confidential Distribution
Provided to GMSL2 Customers Under NDA

42.3.3.6.3 Configuring VTRG when DSC is Active

When DSC is used, the DSC PCLK divide ratio must be specified explicitly as /3. This is because the VTRG is located upstream of the DSC decoder, and as a result the PCLK observed by the VTRG is the PCLK of the compressed video. When DSC is disabled, the default divide ratio of 1 can be used. When using DSC, a fixed value of 0x02 must be written to register 0x7012 (0x02 -> PCLK / 3). Register 0x7012 should only use a value of 0x00 (default, non-DSC) or 0x02 (DSC). Other values are not supported.

In addition, if the HACT period of the video is not integer divisible by 3, then the auto-MSA loading must be disabled (only after an initial MSA load completes to configure all other VTG parameters) and the manual HACT override in registers 0x7047:0x7046 must be programmed with next higher HACT that is integer divisible by 3. For example, if the nominal horizontal resolution is 1600, then HACT is manually programmed to 1602, which is integer divisible by 3.

If auto-VTRG enable is used with DSC, HACT must always be specified regardless of whether HACT is integer divisible by 3. In DSC applications that do not utilize auto-VTRG enable, HACT must only be manually specified when HACT is not integer divisible by 3.

Finally, when DSC is used, the DSC rate control function should be clocked from the VTRG rate control PCLK generator. This ensures the most uniform possible data flow downstream.

All associated registers are detailed in the Configuration Register Details Section

42.3.3.6.4 Additional Configuration Registers (if not using MSA auto-loading)

All VTG parameters can be manually programmed. In most cases, the only VTG parameter that is manually programmed is HACT, and that is only required if DSC is used and HACT/3 is non-integer. All other timing parameters can be automatically loaded via the MSA auto-load from DPTX. It is possible to manually write all of the VTG timing parameters if desired. These can be specified in registers 0x7044 to 0x7053. Refer to the eDP deserializer data sheet for additional details of these registers. In most applications, these registers are not programmed by the user because they typically must match the DP MSA configuration for proper video transmission.

42.3.3.6.5 Sequencing of VTRG Initialization

The VTRG configuration should occur after the DP link is trained. This is required to utilize the MSA auto-load function to configure the VTG from the DPTX (video timing is loaded into the DPTX as part of the DP link training process).

42.3.3.7 VTRG Status Indicators and Debug

The VTRG includes status indicators that can flag disruptions to the data flow.

42.3.3.7.1 Debug Registers

Table 194. VTRG Debug Registers

Register	Register/bit name	Description
0x7008[1:0]	[0] – DE_act_ist [1] – DE_inact_ist	DE activity indicators DE_act_ist is set on each leading edge of DE DE_inact_ist is set on each trailing edge of DE

Analog Devices Confidential, Distribution Only Under NDA
Provided to GMSL2 Customers Under NDA
Maxim Integrated Company Confidential,
ADI Confidential for Review under NDA

		During normal operation, both bits will be continuously set. If they are not set, then the incoming DE may be disrupted
0x7008[3:2]	[2] – VS_act_ist [3] – VS_inact_ist	VS activity indicators VS_act_ist is set on each leading edge of VS VS_inact_ist is set on each trailing edge of VS During normal operation, both bits will be continuously set. If they are not set, then the incoming VS may be disrupted
0x7008[4]	pb_undfl_ist	Pixel Buffer Underflow Indicator 0 – no underflow event (normal operation) 1 – underflow event occurred (problem) An underflow occurs when there is insufficient data in the pixel buffer to satisfy the data flow specified by the downstream rate control PCLK. An underflow condition can occur if there is a sustained difference between the incoming video PCLK and the PCLK generated by the rate control such that the rate control PCLK is higher than the incoming video PCLK. An underflow condition will trigger a reset of the subsystem. In this case, the pixel buffer will begin refilling at the start of the next video frame, and video output will restart when the pixel buffer reaches the nominal fill level.
0x7008[5]	pb_ovfl_ist	Pixel Buffer Overflow Indicator 0 – no overflow event (normal operation) 1 – overflow event occurred (problem) An overflow occurs when there is insufficient space in the pixel buffer to accumulate data arriving from the GMSL2 link video receiver. This can occur if the rate control system pclk output is not correctly tracking the input rate and is outputting data too slowly. Transient changes to the video timing may also be problematic because the rate control loop response is intended to track slow changes, such as PCLK drift. It is not intended to track fast, large transients in the timing. An overflow condition will trigger a reset of the subsystem. In this case, the pixel buffer will begin refilling at the start of the next video frame, and video output will restart when the pixel buffer reaches the nominal fill level.
0x7008[6]	rate_adj_ist	Rate adjust event indicator 0 – No rate adjust event has occurred 1 – Rate adjust event has occurred This bit will typically be set in normal operation when PCLK tracking is enabled, indicating that the control loop is adjusting the rate control PCLK to track the input PCLK. This bit is set every time that the control loop makes a rate adjustment.

ADI Confidential for NEVISION under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customer_Portal, Company Confidential, Under NDA

0x7008[7]	rate_adj_max_ist	Maximum Rate Adjust Event Indicator 0 – No max rate adjust detected (normal operation) 1 – Control loop integral path has reached rate adjust limit If this bit is set, it indicates that the control loop has reached the adjustment limit of the integral control path. Note that the system can still maintain PLK tracking with possibly degraded timing variation based on the control of the proportional control path only. As a result, for modest errors of a couple tenths of a percent beyond the integral control path control range, the system will continue operating more or less normally, although this is not recommended. A max rate adjust condition is typically caused by excessive PCLK error relative to the nominal PCLK.
0x7038[7:0]	PB_CLKP_ADJ_STATUS_B0	Adjusted (actual) PCLK readback in kHz PB_CLKP_ADJ_STATUS_B0 = low byte PB_CLKP_ADJ_STATUS_B1 = middle byte PB_CLKP_ADJ_STATUS_B2 = high bits
0x7039[7:0]	PB_CLKP_ADJ_STATUS_B1	
0x703A[3:0]	PB_CLKP_ADJ_STATUS_B2	

42.3.3.7.2 Interpreting the Status Indicators

If bits 7, 5, or 4 of register 0x7008 are set, then there is a likely problem with the VTRG configuration or the video PCLK. Verify that the actual video PCLK matches the expected value. The VTRG function requires actual PCLK to vary no more than approximately 0.25% from the expected value. Bits 6 and 3:0 of register 0x7008 are typically set during normal operation. If bits 3:0 are not set, it is possible that that a valid video stream is not being received by the GMSL2 input or the VS/DE of this stream are masked in the serializer. All status indicators in register 0x7008 are "write-to-clear." To refresh a status bit, overwrite it with a value of 1. To refresh the entire register, write a value of 0xFF.

42.3.3.7.3 Adjusted PCLK Reporting

The adjusted (actual) PCLK of the video is reported by the VTRG in registers 0x703A:0x7038. This can be a useful debug tool to readily identify PCLK error. In addition, if PCLK adjustments are required elsewhere in the system, the reported value can be used as a convenient means of measuring the absolute PCLK. This value should very precisely represent the actual video PCLK with the primary uncertainty being the crystal frequency reference used by the deserializer.

42.3.3.7.4 Frequency Tolerance

Set the nominal PCLK (PIX_RATE_PER_B[2:0]) to match actual video stream as closely as possible to minimize the PCLK error. The error should be no more than +/- 0.25%. If there is systematic error in the source PCLK, adjust the specified PCLK to match the actual PCLK as closely as possible. As long as the 0.25% tolerance is not violated, the VTRG should lock to the incoming video, and if dynamic DPTX MVID is enabled, the VTRG will then provide a corrected MVID to the sink via the DPTX, ensuring the MVID used by the sink always closely matches the actual PCLK.

Maxim Confidential for New Vision under NDA
Provided to GMSL2 Customers Under NDA

42.3.3.7.5 Measuring HS and VS Uncertainty

One way to observe the impact of the VTRG is to probe the HS and VS signals of the deserializer. They can be routed to GPIO pins for this purpose. With the VTRG disabled, the variation of the period of both HS and VS can be observed with an oscilloscope. With the VTRG enabled, these same measurements can be repeated. This experiment can be conducted for any video timing. Using DSC typically results in the worst timing uncertainty, so DSC should be enabled as a way to create worst case HS and VS uncertainty for a given video timing.
The following table summarizes the typical HS and VS variation observed with and without the VTRG. Note that without the VTRG, uncertainty is typically on the order of 100 pixels while with the VTRG enabled, timing uncertainty is on the order of 1 to 5 pixels. Note that the pseudo PCLK VTRG suppresses both horizontal and vertical timing uncertainty. In contrast, the legacy VTRG mode suppresses horizontal timing uncertainty at the expense of more vertical timing uncertainty.

Table 195. HS and VS Standard Deviations for Two Common Resolutions

Video Resolution	HS Period Stdev VTRG Enabled	HS Period Stdev VTRG Disabled	VS Period Stdev VTRG Enabled	VS Period Stdev VTRG Disabled
1920x1080P60 PCLK=148.5MHz	1.4ns	184ns	5.4ns	136ns
3840x2160P60 PCLK=594MHz	1ns	81ns	6ns	58ns

42.3.3.8 VTRG Video Pattern Generator (VPG)

The VTRG block includes a video pattern generator that can insert a checkerboard test pattern in place of the video stream received from the serializer. A serializer must be connected with active video transmission to enable DP link training when the VPG is used (refer to the VTRG VPG Limitations section for details). However, the VPG generated video pattern, timing, and PCLK are totally independent of the video received from the serializer. The VPG function is only supported by revision 5 (dev_rev=5) and newer devices.

The timing associated with the VPG generated image is automatically generated by the VTG based on the parameters specified during DP link training. The configured PCLK, which should be consistent with MVID and NVID as programmed during link training, is specified in registers 0x7014, 0x7015, and 0x7016 in the same way as during normal VTRG operation. It is important to ensure that MVID and NVID are consistent with the 0x7014/0x7015/0x7016 specified PCLK to ensure robust video recovery in the DP sink device.
The checkerboard pattern size and color component mapping are specified in dedicated registers, which are detailed in the next section. The horizontal and vertical checkerboard size are user programmable, and each 8b RGB color element can be specified as either 0xFF or 0x00 for each alternating checkerboard element, enabling a variety of color patterns to be realized.

42.3.3.8.1 VTRG VPG Registers

The following table summarizes the unique registers associated with the VPG function. Note that some registers are common with standard VTRG video timing regeneration operation, and those registers are detailed in the Configuration Register Details section. Register 0x7013 is specified below and in the primary VTRG register details section because the typical settings are different when using VPG mode.

Table 196. VTRG VPG Configuration Registers

Register	Register/bit name	Description
0x7013[3:0]	PIX_RATE_CLKP_EN PIX_RATE_CLKP_ADJ	0x5 – VPG Mode w/o Dynamic MVID Update to DPTX 0xD – VPG Mode w/ Dynamic MVID Update to DPTX

ADI Confidential for new revision under NDA
Maxim Integrated Confidential, Provided to MSL Customers Under NDA

	VTG_MSA_AUTO_SET DPTX_MCNT_OVRIDE	**Refer to Configuration Register Details for comprehensive description of these bits. Discussed here for clarity in the context of VPG operation**
0x7040[2:0]	VTG_MAN_EN 0x7040[2]	VTG Manual Mode Enable 0 – Manual Mode Disabled (VTG controlled by VTRG controller, normal VTRG operation) 1 – Manual Mode Enabled (VPG mode)
	VTG_PCLK_MODE 0x7040[1:0]	VTG PCLK Mode 0b00 - Horz total/sync counter uses pixel clock Horz active counter uses pixel clock 0b01 - Horz total/sync counter uses pixel clock Horz active counter uses pixel valid **(default)** 0b10 - Horz total/sync counter uses pixel valid Horz active counter uses pixel clock 0b11 - Horz total/sync counter uses pixel valid Horz active counter uses pixel valid **(VPG mode)**
0x7044 **To** **0x7053**	VTG timing parameters	Not typically set by user because MSA auto-loading is used for VTG configuration. VTG parameters typically must match MSAs for correct operation. Auto-load is enabled in register 0x7013.
0x7060	[0] – PIX_PAT_EN	Pattern Generator Enabled. When enabled, generated pattern replaces normal received video stream 0 – Pattern generator disabled (normal operation) 1 – Pattern generator enabled (VPG operation)
	[6:4] - PIX_PAT_COLOR	Pixel pattern color enable. Each bit enables a particular color component for the initial pixel value. Pixel values invert every pixel block. Bit 4 - Red Bit 5 - Green Bit 6 - Blue
0x7064	PIX_PAT_HBLK_LO	Pixel pattern horizontal block size bits 7-0. Horizontal block size is in pixels. The pixel value is inverted every block. 0 – 1 pixel 1 – 2 pixels …….
0x7065	PIX_PAT_HBLK_HI	Pixel pattern horizontal block size bits 7-0. Horizontal block size is in pixels. The pixel value is inverted every block.
0x7066	PIX_PAT_VBLK_LO	Pixel pattern vertical block size bits 7-0. Vertical block size is in lines. The pixel value is inverted every block. 0 – 1 line 1 – 2 lines …….
0x7067	PIX_PAT_VBLK_HI	Pixel pattern vertical block size bits 7-0. Vertical block size is in lines. The pixel value is inverted every block.

ADI Confidential for NEW VISION under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

42.3.3.8.2 VPG Configuration Example

```
# Prerequisites
##########################################
# enable serializer video
# establish video lock
# configure DP link parameters and train link
##########################################
#
# DP link training here
#
# configure VTRG/VPG
# PCLK (should match DP link training; example is PCLK=148.5MHz for 1080P60)
0x90,0x7014,0x14
0x90,0x7015,0x44
0x90,0x7016,0x02
# configure VTRG(no PCLK tracking, MSA auto-load)
0x90,0x7013,0x0D
# manual VTG enable
0x90,0x7040,0x07
# enable pat gen with RGB black/white (all 111s and all 000s alternating)
0x90,0x7060,0x71
# checkerboard size (112 x 112)
0x90,0x7064,0x6F
0x90,0x7065,0x00
0x90,0x7066,0x6F
0x90,0x7067,0x00
# enable vtrg
0x90,0x7000,0x01
```

42.3.3.8.3 VTRG VPG Limitations

4. The DP controller is programmed to check video lock as a prerequisite to training the DP link and enabling DP video transmission. This requires that an upstream serializer with video transmission enabled is connected to the DP deserializer. The serializer to deserializer video transmission is only required to enable the DP controller to observe video lock, and the specific timing and content of the video received from the serializer has no effect whatsoever on the video pattern or timing that is generated by the VPG.

5. The DP controller will automatically configure some VTRG settings when used. As a result, it is recommended to configure and enable the VPG after link training to ensure all configuration is as desired and nothing has been unexpectedly changed by the DP controller.

6. The timing of the video generated by the VTG must match the MSA configuration of the DP output. In most cases, this is handled automatically by the auto-MSA load function, although the user can manually configure all timing parameters in the VTRG. This is not truly a limitation of the VPG. Rather, it is simply mandated by the requirement that the DP video stream timing must match that of the DP MSA configuration. MSA information must be configured via link training prior MSA information being automatically loaded into VTRG if the automatic load function is utilized.

API Confidential for NewVision under NDA
Analog Devices Confidential - Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL Customers Under NDA

42.3.3.9 Legacy VTRG Configuration (required for rev. 4 and older devices)

Legacy VTRG operation must be used on all devices prior to revision 5. Newer devices do support the legacy operating mode, but in the case of newer devices it should only be used in situations where backward compatibility with legacy software must be maintained.

42.3.3.9.1 Legacy VTRG Limitations

Legacy VTRG operation has some notable limitations when compared to the pseudo PCLK VTRG:

3. Pseudo PCLK VTRG suppresses both horizontal and vertical timing uncertainty while legacy VTRG suppresses horizontal timing uncertainty at the expense of increased vertical timing uncertainty.
4. Horizontal Resolution Limitation with DSC
   Legacy VTRG functionality in device revision 4 (DEV_REV=4) and older have the following limitation. Newer devices utilizing the legacy VTRG function do not have this limitation.

$$roundup\left(\frac{H_{active}}{3}\right) = even$$

42.3.3.9.2 Legacy VTRG Configuration Details

There are four steps to program the VTRG:

9. Setup pixel rate control
10. Setup video timing generator (VTG)
11. Adjust control loop parameters
12. Enable VTRG

Steps 1–3 require registers to be programmed with the exact video timings for the given use case. There is a tool in the GMSL GUI that automatically generates the required configuration for steps 1-4 above. The following table detail all registers that are written.

Table 197. Programming Steps to Enable VTRG

Register Address	Bitfield	Value	Register Description
0x4507	pix_rate_ext_en[0]	Set high to enable	Enables external control. If using DSC, must be set after loading DSC profile
0x7010	pix_rate_clk_act[4:0]	Program with "Pix rate clks/act reg value" from c-code (in hex)	Pixel rate clocks per active cycle
0x7010	pix_rate_ctrl_en[0]	set bit 7 high (don't overwrite pixel rate clks/act)	Pixel rate control enable
0x7011	pix_rate_inact_red[7:0]	Program with "Pix rate red reg value" from c-code (in hex)	Pixel rate inactive-to-inactive redistribution count
0x7012	pix_rate_div[1:0]	Set to 0x00 for no DSC set to 0x02 if using DSC	Pixel rate divide factor to match sparse pixel rates, if DSC compression enabled divide by 3, else divide by 1

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential, Distribute only under NDA
Provided to GMSL2 Customers Under NDA

0x7014 0x7015	pix_rate_act_lo[7:0] pix_rate_act_hi[1:0]	Program with "Pix rate active reg value" from c-code (in hex)	Pixel rate active count bits 7–0
0x7016 0x7017	pix_rate_inact_lo[7:0] pix_rate_inact_hi[2:0]	Program with "Pix rate inactive reg value" from c-code (in hex)	Pixel rate active count bits 7–0
0x7044 0x7045	vtg_htot_lo[7:0] vtg_htot_hi[7:0]	Program with "Horiz total" from c-code (in hex)	VTG horizontal total (line length) bits 7–0, programmed in numbers of 600Mhz clocks
0x7046 0x7047	vtg_hact_lo[7:0] vtg_hact_hi[7:0]	Program with actual Hactive in pixels (divide by 3 is DSC is used)	VTG horizontal active bits 7–0, programmed in numbers of pixels or compressed bytes (pix/3)
0x7048 0x7049	vtg_hstt_lo[7:0] vtg_hstt_hi[7:0]	Program with "Horiz start" from c-code (in hex)	VTG horizontal sync start (sw + bp) bits 7–0, programmed in numbers of 600Mhz clocks
0x704A 0x704B	vtg_hsw_lo[7:0] vtg_hsw_hi[7:0]	program with "Hsync width" from c-code. Set MSB for positive sync polarity	VTG horizontal sync width bits 7–0, programmed in numbers of 600Mhz clocks. MSB is sync polarity.
0x704C 0x704D	vtg_vtot_lo[7:0] vtg_vtot_hi[7:0]	Program with Vtotal in lines	VTG vertical total (frame length) bits 7–0, programmed in numbers of horizontal lines
0x704E 0x704F	vtg_vact_lo[7:0] vtg_vact_hi[7:0]	Program with Vactive in lines	VTG vertical active bits 7–0, programmed in numbers of horizontal lines
0x7050 0x7051	vtg_vstt_lo[7:0] vtg_vstt_hi[7:0]	Program with Vsw + Vbp (sync width plus back porch)	VTG vertical sync start (sw + bp) bits 7–0, programmed in numbers of horizontal lines
0x7052 0x7053	vtg_vsw_lo[7:0] vtg_vsw_hi[7:0]	Program with Vsw. Set MSB for positive sync polarity	VTG vertical sync width bits 7–0, programmed in numbers of horizontal lines
	Additional VTRG settings to optimize performance	Set following registers 0x7020 -> 0x07 0x702D -> 0x01 0x702E -> 0x01 0x702F -> 0x01 0x7030 -> 0x04	Additional VTRG settings to optimize performance
0x7000	vtrg_en[0]	0x01	Video Timing Regenerator (VTRG) enable

ADI Confidential for Newvision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

42.3.3.9.2.1 Example VTRG Scripts

Two Legacy VTRG example scripts are provided for VESA standard 1080p with and without DSC compression enabled.

42.3.3.9.2.1.1 Example Script 1 – 1080p with no DSC

```
# Example VTRG Script for no DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 148.50
# Htotal = 2200
# Hactive = 1920
# Hfp = 88
# Hsw = 44
# Hbp = 148
# Vtotal = 1125
# Vactive = 1080
# Vfp = 4
# Vsw = 5
# Vbp = 36
#
# Enable external pixel rate control
0x90,0x4507,0x80
# Pixel Rate Clks/Act - program to "Pix rate clk/act" from c-code & set MSB to enable
0x90,0x7010,0x83
# Pixel Rate Redist – program with “Pix rate red Reg value” from c-code
0x90,0x7011,0x00
# Pixel Rate Divide - program to 0x00 if no DSC, program to 0x02 if DSC used
0x90,0x7012,0x00
# Pixel Rate Active - program with "Pix rate active Reg value" from c-code
0x90,0x7014,0x62
0x90,0x7015,0x00
# Pixel Rate inactive - program with "Pix rate active Reg value" from c-code
0x90,0x7016,0x2C
0x90,0x7017,0x01
#
# VTG Horizontal Total - program with "Horiz total" from c-code
0x90,0x7044,0x9B
0x90,0x7045,0x22
# VTG Horizontal Active - program with Hactive (divide by 3 is DSC used)
0x90,0x7046,0x80
0x90,0x7047,0x07
# VTG Horizontal Sync Start - program with "Horiz start" from c-code
0x90,0x7048,0x0A
0x90,0x7049,0x03
# VTG Horizontal Sync Width & Polarity - program with "Hsync width" from c-code.
# Set MSB for positive sync polarity
0x90,0x704A,0x58
0x90,0x704B,0x80
# VTG Vertical Total - Program with Vtotal in lines
0x90,0x704C,0x65
0x90,0x704D,0x04
# VTG Vertical Active - Program with Vactive in lines
0x90,0x704E,0x38
0x90,0x704F,0x04
# VTG Vertical Sync Start - Program with Vsw + Vbp
0x90,0x7050,0x29
0x90,0x7051,0x00
# VTG Vertical Sync Width - Program with Vsw
# Set MSB for positive sync polarity
0x90,0x7052,0x05
```

Maxim Integrated Confidential Distribute only under NDA
Provided to GMSL2_Customers Under NDA
Maxim Confidential for New Vision under NDA

```
0x90,0x7053,0x80
#
# Additional VTRG settings (optimizations)
0x90,0x7020,0x07
0x90,0x702D,0x01
0x90,0x702E,0x01
0x90,0x702F,0x01
0x90,0x7030,0x04
#
# VTRG Enable
0x90,0x7000,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential- Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.3.3.9.2.1.2 Example Script 2 – 1080p with DSC

```
# Example VTRG Script for using DSC
# NC 29jan20
#
# Video timing - VESA standard 1080p
# PCLK = 148.50
# Htotal = 2200
# Hactive = 1920
# Hfp = 88
# Hsw = 44
# Hbp = 148
# Vtotal = 1125
# Vactive = 1080
# Vfp = 4
# Vsw = 5
# Vbp = 36
#
# Enable external pixel rate control
0x90,0x4507,0x80
# Pixel Rate Clks/Act - program to "Pix rate clk/act" from c-code & set MSB to enable
0x90,0x7010,0x83
# Pixel Rate Redist – program with “Pix rate red Reg value” from c-code
0x90,0x7011,0x00
# Pixel Rate Divide - program to 0x00 if no DSC, program to 0x02 if DSC used
0x90,0x7012,0x02
# Pixel Rate Active - program with "Pix rate active Reg value" from c-code
0x90,0x7014,0x62
0x90,0x7015,0x00
# Pixel Rate inactive - program with "Pix rate active Reg value" from c-code
0x90,0x7016,0x2C
0x90,0x7017,0x01
#
# VTG Horizontal Total - program with "Horiz total" from c-code
0x90,0x7044,0x9B
0x90,0x7045,0x22
# VTG Horizontal Active - program with Hactive (divide by 3 is DSC used)
0x90,0x7046,0x80
0x90,0x7047,0x02
# VTG Horizontal Sync Start - program with "Horiz start" from c-code
0x90,0x7048,0x0A
0x90,0x7049,0x03
# VTG Horizontal Sync Width & Polarity - program with "Hsync width" from c-code.
# Set MSB for positive sync polarity
0x90,0x704A,0x58
0x90,0x704B,0x80
# VTG Vertical Total - Program with Vtotal in lines
0x90,0x704C,0x65
0x90,0x704D,0x04
# VTG Vertical Active - Program with Vactive in lines
0x90,0x704E,0x38
0x90,0x704F,0x04
# VTG Vertical Sync Start - Program with Vsw + Vbp
0x90,0x7050,0x29
0x90,0x7051,0x00
# VTG Vertical Sync Width - Program with Vsw
# Set MSB for positive sync polarity
0x90,0x7052,0x05
0x90,0x7053,0x80
#
# Additional VTRG settings (optimizations)
0x90,0x7020,0x07
0x90,0x702D,0x01
```

Maxim Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
Maxim Confidential - Distribution only under NDA
Confidential for Newvision under NDA

```
0x90,0x702E,0x01
0x90,0x702F,0x01
0x90,0x7030,0x04
#
# VTRG Enable
0x90,0x7000,0x01
#
# Debug - HS Out Enable on GPIO00, VS Out Enable on GPIO01
0x90,0x01DF,0x06
```

42.3.3.9.3 Legacy VTRG Debug

The primary indicator that the VTRG is operating correctly is by checking the pixel buffer underflow/overflow interrupt statuses, pb_undfl_ist and pb_ovfl_ist (register 0x7008)***Error! Reference source not found.***. Since these are latching bits and may have been inadvertently set at startup, first clear the bits by writing them to 1. Then, read the bits to check for underflow/overflow interrupts. Underflow or overflow may occur if:

- VTG configuration does not match the original video mode timings.
  - Verify that the VTG settings match the original video mode timings. In particular, check the calculated values for horizontal total, horizontal start, and horizontal sync width.
- Pixel rate control configuration does not match the original video nominal pixel clock.
  - Use the calculator tool to verify settings.
  - Check that pix_rate_div is correct for DSC/non-DSC video.
- Pixel buffer control loop adjustments are too tight for the original video mode (i.e., pb_rate_adj_max is set too low for the allowed original video pixel clock PPM offset from nominal).
  - Check the rate_adj_max_ist interrupt status. This is a useful indicator that adjustments have reached the maximum limit. It is not abnormal for it to be set, but if it is not set that indicates that there is additional margin available on the rate adjustment.
  - Repeatedly read the live rate adjustment status, vtg_ht_adj. This live bit indicates if the rate adjustment is centered or skewed to the negative or positive side.
  - Read the rate adjustment min/max watermark, vtg_ht_adj_min/max. These provide information regarding the minimum and maximum excursions for rate adjustment, especially for the very wide default setting for pb_rate_adj_max. Wide excursions may indicate that the gain settings (pb_rate_adj_step_* settings) are too high and causing instability in the control loop. It is also an indication of the jitter on the incoming video.

Table 198. Legacy VTRG Debug Registers

Register Address	Bitfield	Description
0x7000	vtrg_en[0]	Enables VTRG operation
0x7008	pb_undfl_ist[0]	This latching bit indicates that the pixel buffer underflowed. It must be initially cleared (write 1 to clear) after VTRG enable. It will remain cleared when the VTRG is operating properly.
0x7008	pb_ovfl_ist[0]	This latching bit indicates that the pixel buffer overflowed. It must be initially cleared (write 1 to clear) after VTRG enable. It will remain cleared when the VTRG is operating properly.
0x7008	rate_adj_ist[0]	This latching bit indicates that a rate adjust event occurred. It must be initially cleared (write 1 to clear) after VTRG enable. It is normal to be set when the VTRG is operating properly.

ADI Confidential Information under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customer_Name NDA

0x7008	rate_adj_max_ist[0]	This latching bit indicates that a rate adjust event was limited by the maximum setting (pb_rate_adj_max). It must be initially cleared (write 1 to clear) after VTRG enable. It is not abnormal to be set when the VTRG is operating properly but does warn that the adjustment limit is being reached.
0x7010	pix_rate_ctrl_en[0]	Enables pixel rate control that provides a pixel data valid to control the pixel flow rate associated with the video pipeline clock. The pixel data valid paces pixel output at the nominal pixel clock rate.
0x7020	pb_del_acc_per[3:0]	Sets the period (number of horizontal lines) for accumulating the pixel buffer fill delta information used to make rate adjustments. This value is the zero-relative number of lines. This is the control loop bandwidth.
0x702C	pb_rate_adj_step_0[5:0]	Sets the rate adjust step size for range 0. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 0.
0x702D	pb_rate_adj_step_1[5:0]	Sets the rate adjust step size for range 1. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 1.
0x702E	pb_rate_adj_step_2[5:0]	Sets the rate adjust step size for range 2. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 2.
0x702F	pb_rate_adj_step_3[5:0]	Sets the rate adjust step size for range 3. Each incremental step size changes the horizontal line length by 1 video pipeline clock (1.6667ns). This is the control loop gain for range 3.
0x7030	pb_rate_adj_max[6:0]	Sets the maximum allowed rate adjustment. This 2's complement positive value is used as a +/- limit. A value of zero disables the limit. This limits the ppm offset of the pixel clock with respect to the video pipeline clock that can be tracked. It also limits the magnitude of jitter transfer of the input video to output video.
0x7035	vtg_ht_adj[7:0]	VTG horizontal total adjust live status. This is a 2's-complement value.
0x7036	vtg_ht_adj_min[7:0]	VTG horizontal total adjust minimum watermark. This captures and holds the minimum (most negative) horizontal total adjust since the last reset. Writing (any value) will reset the value to zero. This is a 2's-complement value.
0x7037	vtg_ht_adj_max[7:0]	VTG horizontal total adjust maximum watermark. This captures and holds the maximum (most positive) horizontal total adjust since the last reset. Writing (any value) will reset the value to zero. This is a 2's-complement value.

ADI Confidential for New Vision under NDA
Analog Devices Inc. Maxim Integrated Confidential Provided to GMSL2_Customer_Under_NDA

42.4 Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

42.4.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

ADI Confidential for New Version under NDA
Analog Devices Confidential Only
Maxim Integrated Company Confidential
Provided to GMSL Customers Under NDA

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock, lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

42.4.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the

ADI Confidential for New Vision under NDA
Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA

		DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

42.5 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43 CSI-2 Serializers and Deserializers

43.1 Overview

This section describes the architecture and programming of GMSL2 CSI-2 D-PHY serializers and deserializers. GMSL2 CSI-2 (Camera Serial Interface) devices bridge MIPI CSI-2 input and output interfaces via a serial link connection. The MIPI D-PHY ports have configurable lane arrangements and data routing options that enable flexible system configurations. Reference Figure 177 for an example GMSL2 CSI-2 system block diagram.

Note: The DSI and CSI-2 packet-level protocols are different and not interchangeable. A CSI-2 source must be connected to a CSI-2 serializer (CSI-2 sink); a DSI source must be connected to a DSI serializer (DSI sink). Any GMSL2 serializer can connect to a GMSL2 CSI-2 deserializer. A CSI-2 deserializer can be connected to any CSI-2 sink device (typically an SoC). Reference the DSI Serializers section for more information.

This section covers programming and configuration of all GMSL2 CSI-2 device families except GMSL2 CSI-2 Quad Deserializers (see *CSI-2 Quad Deserializers* section for more information). Reference Feature Availability by Device Family for a complete list of GMSL2 CSI-2 devices.

Note: Reference the *GMSL1 Legacy Mode (Backward Compatibility)* section for configuration details for GMSL1 Mode and data.

GMSL2 CSI-2 devices include the following features and functions:

- Video Crossbar
- GMSL1 backward compatibility
- Individually configurable lane rates (80Mbps–2.5Gbps per lane)
- Data Formats: RAW8/10/12/14/16/20, RGB565/666/888, YUV422 8/10-bit, and User-Defined and Generic Long-Packet Data Types
- Configurable GMSL2 link rates
  - 3Gbps or 6Gbps Forward Link Rates
  - 187.5Mbps Reverse Link Rate
- Refer to device-specific datasheets for full feature lists

GMSL2 CSI-2 Serializers:

- MIPI CSI-2 v1.3 input port with virtual channel support
- Advanced MIPI D-PHY v1.2 receivers with polarity flip and data-lane reassignment

GMSL2 CSI-2 Deserializers:

- MIPI CSI-2 v1.3 output port with virtual channel support
- Advanced MIPI D-PHY v1.2 transmitters with polarity flip and data-lane reassignment

Note: For information regarding the CSI-2 Quad Deserializers, reference the *CSI-2 Quad Deserializers* section.

ADI Confidential for Newvision under NDA
Analog Confidential Distribution Only
Maxim Integrated Company Confidential,
Provided to GMSL_2_Customers Under NDA

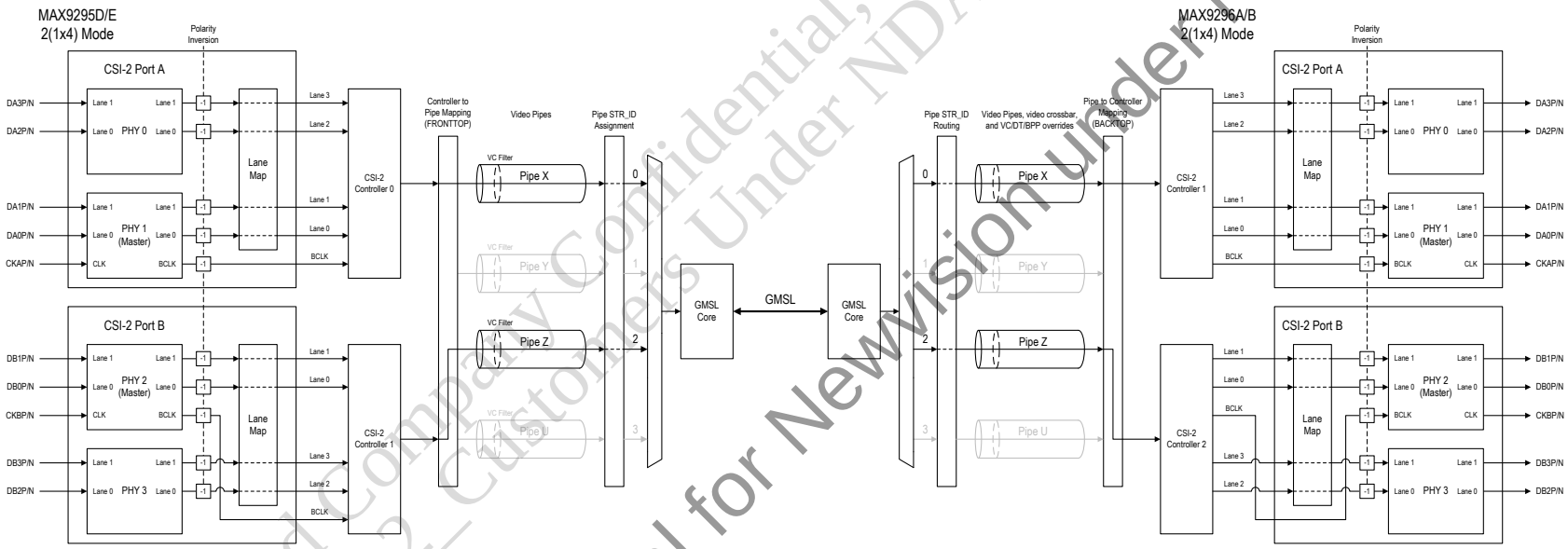


Figure 177. Example GMSL2 CSI-2 System Block Diagram

Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
ADI Confidential for New Revision under NDA

43.2 Operation

43.2.1 GMSL2 CSI-2 Serializers

GMSL2 CSI-2 serializers convert MIPI D-PHY v1.2 input to GMSL1 or GMSL2 serial link output. Differential CSI-2 data is input to the MIPI D-PHY receiver block(s) within the MIPI receive ports. Note that the specific architecture of the MIPI receive ports varies by device and configuration, and the MIPI D-PHY receiver block(s) have programmable lane counts. In all arrangements, the MIPI receive ports process the CSI-2 video input and provide configuration options for the clock and data lanes. Each clock and data lane supports polarity inversion; data lanes also support deskew calibration to correct timing with high-speed transfer rates. A programmable lane map allows data lane reassignment within the MIPI port to provide flexibility with PCB layout and routing.

CSI-2 protocol consists of:

- **Long packets** – Application-specific payload data.
- **Short packets** – Frame and line synchronization data and other image-related parameters.

The MIPI receive port outputs to the MIPI CSI-2 v1.3 Rx controller. The controller converts the CSI-2 video stream into generic parallel 8- to 24-bit video data with sync signals. This parallel data is routed to the video pipes (with configurable pipe assignments via register writes). The video pipe output streams are assigned stream IDs before being processed at the GMSL Tx PHY.

Note: Pipe X is the only video pipe that supports watermark generation.

A block diagram of the MAX9295D/E GMSL2 CSI-2 serializer (Figure 178) shows the MIPI ports, CSI-2 controllers, video pipe routing, and GMSL core.

ADI Confidential for Review under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

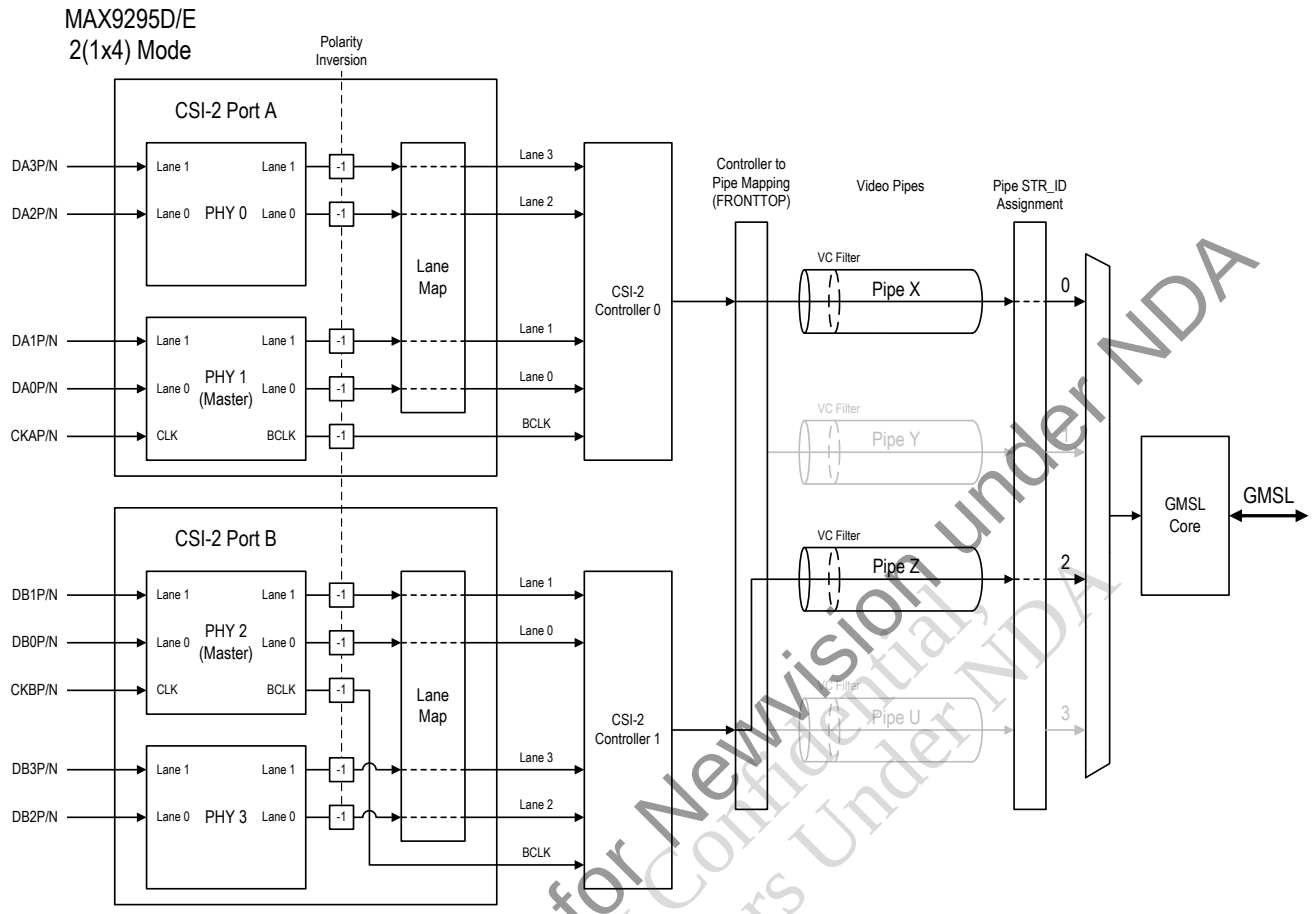


Figure 178. MAX9295D/E CSI-2 Serializer Block Diagram

43.2.1.1 MIPI D-PHY Rx

The GMSL2 CSI-2 serializer device variants share similar MIPI D-PHY v1.2 architecture. There are four total MIPI PHYs (i.e., PHY0–3). The PHY availability is dependent on the part number/package and the phy_config bitfield settings. Each PHY supports two differential data lanes, but clock input is only supported by PHY1 and PHY2.

The PHYs are grouped to form the MIPI port(s). Each port has one clock lane, though the number of data lanes is configurable. When multiple PHYs are combined to form a single port, the PHY that receives the MIPI clock is referred to as the "master" PHY.

Port notation uses the following convention: number of ports (number of clock lanes x number of data lanes). For example, a serializer with two ports that each contain two data lanes is expressed as 2(1x2). This can be further condensed to 2x2. The number of data lanes in this notation is the maximum available, but the ports can be optionally configured to use less than the maximum. Four-lane ports can operate using one, two, or four data lanes; two-lane ports can operate with one or two data lanes.

Analog Devices Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

43.2.1.2 MIPI CSI-2 Rx Controller

There are two MIPI Rx Controllers in each serializer: Controller 0 processes the video packets from Port A and Controller 1 processes the video packets from Port B. For serializers with only one MIPI port (MAX9295A), only Controller 1 is used. Reference the *CSI-2 Serializer Port Architecture* section for port-to-controller mappings. The MIPI CSI-2 controller block diagram is shown in *Figure 179*.

The MIPI CSI-2 controller contains the following functional blocks:

- **Lane Merger**: Receives the incoming data from multiple D-PHY data lanes and merges the data to a single packet.
- **Packet Processor**: Receives the merged packet and extracts the header, footer, payload data, and sync signals.
- **ECC Check / Correct**: The integrity of the packet header ECC is checked. Any errors detected can be reported to ERRB.
- **CRC Check**: Checks for payload CRC errors and outputs error status to ERRB.
- **Video Formation**: The video signals are output from the MIPI controller and routed to the video pipes.

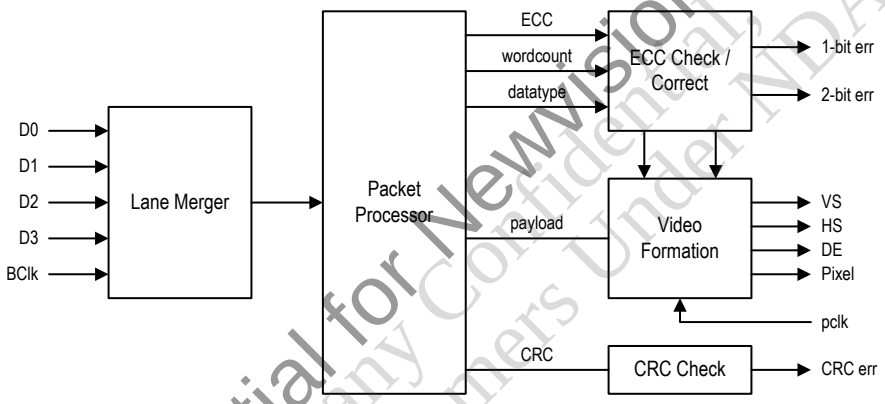


Figure 179. MIPI CSI-2 Controller Block Diagram

The internal video pipe PCLK can be calculated from the MIPI input using the equation below:

$$PCLK = \frac{MIPI\ Lane\ Rate * Lane\ Count}{bits\ per\ pixel}$$

Note: Reference the GMSL2 Interface-Specific Bandwidth Calculations section for additional discussion of the PCLK and its relationship to interface and serial link bandwidth.

The MIPI Rx controllers output to the video pipes. Controller-to-Pipe Mapping is configurable via register writes. Note that video pipes have only one input.

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

43.2.1.2.1 CSI-2 Packets

The packet header is defined by the MIPI CSI-2 specification and includes the packet data type, virtual channel, word count, and Error Correction Code (ECC). The MIPI controller incorporates an ECC Check/Correct block to determine the integrity of the packet header. An ECC error indicates the presence of a bit error in the packet header. A packet header error can cause the output video to have an unexpected data type, unexpected line length, or completely corrupted video data. In the case of a 1-bit ECC error, the MIPI controller automatically corrects the header. With a 2-bit ECC error, the controller will assert the ECC error bit and drop the entire packet.

The packet footer consists of a 16-bit CRC calculated from the payload (pixel) data. The CRC check block calculates the CRC of the received payload and compares it with the received CRC in the packet footer. If the received and calculated CRC do not match, the controller will assert the CRC error bit indicating that the video data may be corrupted.

Each controller contains a video formation block that generates the video data and sync signals from the CSI-2 packets (Figure 180). Unpacking the CSI-2 packets and converting the video to a standard format allows interoperability between GMSL2 devices. The vertical sync (VS) signal is generated from CSI-2 Frame Start (FS) and Frame End (FE) packets: the FS packet sets VS high, while the FE packet sets VS low. CSI-2 protocol does not require a horizontal sync (HS) signal, but optional Line Start (LS) and Line End (LE) packets may be used. The video formation block uses LS and LE signals, if present, to generate the HS signal; if the signals are not present, the HS signal in the generated video stays low. The Data Enable (DE) signal is derived from the length of the long packets (each of which contains a line of active pixels). The bits per pixel (bpp) rate is determined by the data type specified in the header of the CSI-2 long packet.

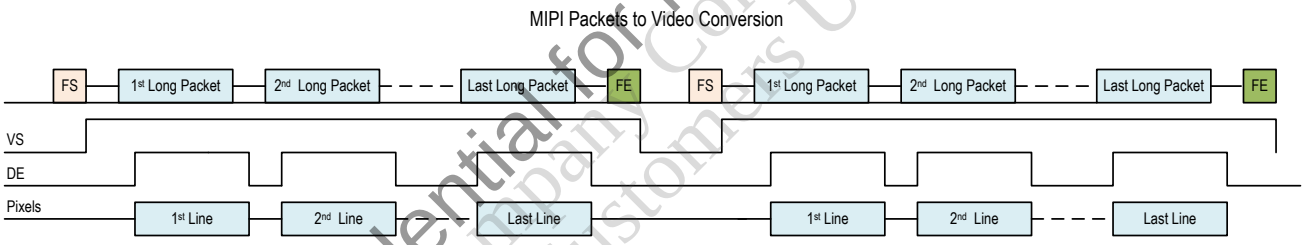


Figure 180: MIPI CSI-2 Packets to Video Conversion (Two Frames)

Note: The DE and Pixel signals must be processed and stored within internal memory before being transmitted over the GMSL link. This processing and storage results in a lag compared to the incoming CSI-2 signals. VS signals do not require memory and do not have latency.

CSI-2 can support multiple bpp rates on a single CSI-2 bit clock (i.e., variable PCLK). To accommodate this, CSI-2 serializers will only generate an internal PCLK and assert PCLKDET if both a valid CSI-2 clock and valid long packets (pixel data) have been received. A PCLK will not be generated from a CSI-2 bit clock alone.

ADI Confidential for New Vision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

43.2.2 GMSL2 CSI-2 Deserializers

GMSL2 CSI-2 deserializers operate in the opposite direction from CSI-2 serializers. The functional blocks in the deserializers operate similarly but are arranged in a mirrored layout.

GMSL2 CSI-2 deserializers convert GMSL1 or GMSL2 serial link input to MIPI D-PHY v1.2 output. Serial link input is received and processed at the GMSL Rx PHY. Standard format video data is then routed to any deserializer video pipe based on the stream ID which is set in the serializer. Multiple deserializer video pipes may share the same stream ID.

Note: Pipe X is the only video pipe that supports watermark detection.

The video pipes output to the MIPI CSI-2 v1.3 Tx controllers. The Pipe-to-Controller Mapping is user-configurable via register writes. Note that each video pipe has only one output, but multiple video pipes can connect to a single controller via FCFS aggregation. All GMSL2 CSI-2 deserializers feature four controllers. The controller converts the standard video data input to CSI-2 video output. Controller-to-port mapping is fixed.

The video data output from the CSI-2 controller(s) is routed to the MIPI D-PHY transmit block(s) within the MIPI transmit ports. Note that the specific architecture of the MIPI transmit ports varies by device package and configuration, and the MIPI D-PHY transmitter block(s) have configurable lane counts. In all arrangements, the MIPI transmit ports process the CSI-2 video from the controllers and provide configuration options for the clock and data lanes. Each clock and data lane supports polarity inversion. A programmable lane map allows data lane reassignment within the MIPI port to provide flexibility with PCB layout and routing.

A block diagram of the MAX9296B GMSL2 CSI-2 deserializer (Figure 181) shows the GMSL core, video pipe routing, CSI-2 controllers, and MIPI ports.

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

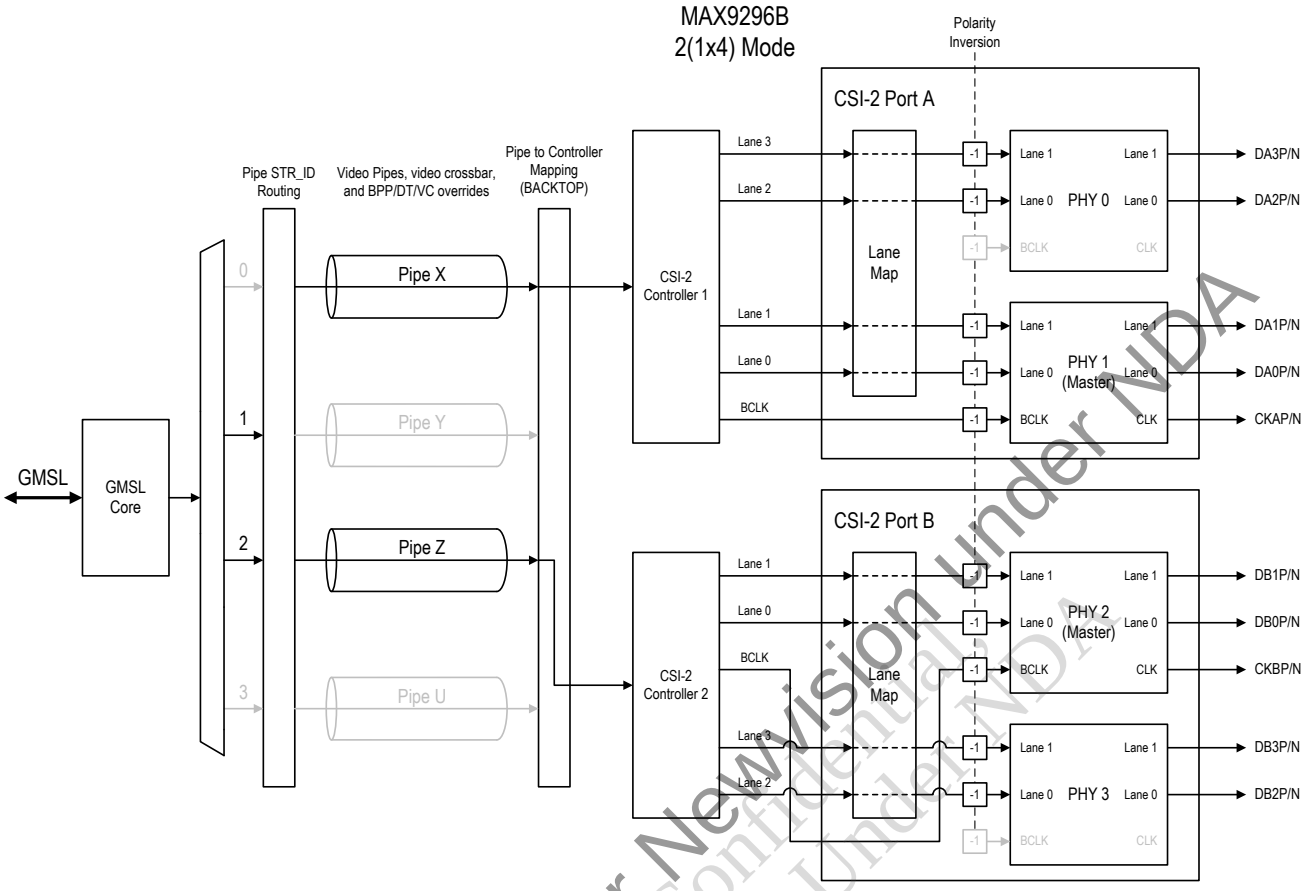


Figure 181. MAX9296B CSI-2 Deserializer Block Diagram

43.2.2.1 MIPI CSI-2 Tx Controller

There are four MIPI Tx Controllers in each GMSL2 CSI-2 deserializer. The controller receives standard format video data from the video pipe and converts it into the MIPI CSI-2 format. A video pipe can only be connected to a single MIPI controller, but multiple pipes can connect to a single controller via FCFS aggregation (see Aggregation). The controller-to-port mapping is fixed (arrangement is dependent on part number, package, and phy_config bitfield). All routing must be performed with video pipe-to-controller mapping. The MIPI transmit port(s) to be used for the deserializer output will determine the required video pipe-to-controller mapping. Reference the *CSI-2 Deserializer Port Architecture* section to determine which controller is connected to the desired MIPI output port.

The MIPI Tx controller (deserializer) performs the reverse function of the MIPI Rx controller (serializer): video data and sync signals are combined with the CSI-2 header information (i.e., data type and VC) to form CSI-2 packets. ECC and CRC codes are generated. The number of MIPI data lanes is defined with the CSI2_LANE_CNT bitfield.

Note: The CSI-2 CRC is generated from the data that enters MIPI Tx controller. Upstream bit errors will not be detected in the MIPI CRC output from the deserializer.

ADI Confidential for Newvision Under NDA
Maxim Integrated Company Confidential
Provided to MSI Customers Under NDA

43.2.2.2 MIPI D-PHY Tx

The GMSL2 CSI-2 deserializer device variants share similar MIPI D-PHY v1.2 architecture. There are four MIPI PHYs (i.e., PHY0–3) in total. The PHY availability is dependent on the part number/package and the phy_config bitfield settings. For most devices, each PHY supports two differential data lanes and one differential clock output. Devices with two four-lane outputs only support clock output on PHY1 and PHY2 (MAX9296A).

The PHYs are grouped to form the MIPI port(s). Each port has one clock lane, though the number of data lanes is configurable. When multiple PHYs are combined to form a single port, the PHY that outputs the MIPI clock is referred to as the “master” PHY.

Port notation uses the following convention: number of ports (number of clock lanes x number of data lanes). For example, a deserializer with two ports that each contain two data lanes is expressed as 2(1x2). This can be further condensed to 2x2. The number of data lanes in this notation is the maximum available, but the ports can be optionally configured to use less than the maximum. Four lane ports can operate using one, two, or four data lanes; two lane ports can operate with one or two data lanes.

Note: Each MIPI port supports one clock lane. When multiple PHYs are combined to form a single port (i.e., four-lane mode), the PHY that outputs the MIPI clock is referred to as the “master” PHY. In GMSL2 CSI-2 deserializer four-lane mode, master PHY support is restricted to PHY1 or PHY2 (depending on device configuration.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only to Maxim Integrated Company Confidential Provided to GMSL2_Customers Under NDA

43.3 Serializer Configuration

The GMSL2 CSI-2 serializer must be enabled and configured before the MIPI source begins transmitting. Dynamic configuration is not supported. In accordance with the CSI-2 standard, the Tx clock and data lanes require special start and stop patterns to set the CSI-2 receiver into HS/LP mode.

To power-up the GMSL2 CSI-2 serializer and the GMSL system, the serializer must be reset/enabled and the GMSL system configured while the CSI-2 source is disabled.

The GMSL2 CSI-2 serializer is configured in the following order:

1. Configure the MIPI D-PHY Rx.
2. Configure MIPI port lane mapping and polarity inversion (if necessary).
3. Adjust deskew and PHY timing settings.
4. Program video pipes and routing to GMSL core.

Following the successful configuration of the GMSL system, the CSI-2 source/imager sensor can be enabled.

Note: GMSL2 CSI-2 serializer packages have different CSI-2 MIPI port options and configurations. Other GMSL2 features may vary between part number variants. See Table 202 for GMSL2 CSI-2 serializer device comparisons.

Table 202. GMSL2 CSI-2 Serializer Port Comparison

Part Number	Package Size	Number of MIPI Ports	Data Lanes per Port	Number of GMSL Links
MAX9295A	5mm x 5mm 32-Pin TQFN	1	1, 2, or 4	1
MAX9295B	5mm x 5mm 32-Pin TQFN	2	1 or 2	1
		1	1, 2, or 4	1
MAX9295D	7mm x 7mm 48-Pin TQFN	2	1, 2, or 4	1
MAX9295E	8mm x 8mm 56-Pin TQFN	2	1, 2, or 4	2

ADI Confidential for New Vision under NDA
Analog Devices Internal Use Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

43.3.1 D-PHY Rx Configuration

The PHYs are grouped to form the MIPI port(s). Each port must have one (active) clock lane, though the number of data lanes is configurable. When multiple PHYs are combined to form a single port, the PHY that outputs the MIPI clock is referred to as the “master” PHY. Each PHY supports two differential data lines, but only PHY1 and PHY2 can receive clock input.

The MIPI D-PHY bitrate is auto-detected by the GMSL2 CSI-2 serializer Rx PHY and does not need to be pre-programmed.

Description of MIPI port modes:

- **1x4 Mode**: PHY1 and PHY2 are combined to form a single four-lane port with the clock received on PHY2 (Figure 182 and Figure 183).
- **2x2 Mode**: PHY1 and PHY2 are used as two independent two-lane ports with each port receiving a clock (Figure 184).
- **2x4 Mode**: Two independent four-lane ports are used. PHY 0 and PHY1 are combined to form a four-lane port with the clock received on PHY 1. PHY2 and PHY3 are combined to form a four-lane port with the clock received on PHY2 (Figure 185).

The CSI-2 serializer variants have specific MIPI Port architectures. The port mode is set using the phy_config register; lane counts are set with the ctrl0_num_lanes and ctrl1_num_lanes registers. Refer to *Table 203* for the PHYs and MIPI port modes available to each part number. Configuration registers are presented in *Table 204*.

Table 203. CSI-2 Serializers: MIPI Ports by Part Number

Part Number	PHYs Available	MIPI Port Modes
MAX9295A	PHY1 and PHY2	1x4
MAX9295B	PHY1 and PHY2	1x4 and 2x2
MAX9295D/E	PHY0–3	1x4 and 2x4

ADI Confidential for New Version under NDA
Analog Devices Confidential, Property of Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Table 204. GMSL2 CSI-2 Serializer PHY Configuration Registers

Register Address	Bitfield Name	Bits	Part Number	POR	Decode
0x330	phy_config[2:0]	2:0	MAX9295A	0b000	0b000: 1x4
			MAX9295B	0b011	0b000: 1x4 0b001: 2x2, Port A only (1x2) 0b010: 2x2, Port B only (1x2) 0b011: 2x2, Port A and B
			MAX9295D	0b110	0b100: 2x4, Port A only (1x4) 0b101: 2x4, Port B only (1x4) 0b110: 2x4, Port A and B
			MAX9295E	0b100	0b100: 2x4, Port A only (1x4) 0b101: 2x4, Port B only (1x4) 0b110: 2x4, Port A and B
			MAX9295A/B/D/E		0b111: RSVD
0x330	CSI_ENABLE	7	MAX9295E	0b0	0b0: CSI disabled 0b1: CSI enabled Note: This bit only exists on MAX9295E. It must be set high to enable the CSI-2 input.
0x331	ctrl0_num_lanes[1:0]	1:0	MAX9295D/E	0b11	Number of data lanes used with controller 0: 0b00: One data lane 0b01: Two data lanes 0b10: RSVD 0b11: Four data lanes
			MAX9295B	0b01	
0x331	ctrl1_num_lanes[1:0]	5:4	MAX9295A/D/E	0b11	Number of data lanes used with controller 1: 0b00: One data lane 0b01: Two data lanes 0b10: RSVD 0b11: Four data lanes
			MAX9295B	0b01	

Note: MIPI port modes indicate the maximum number of lanes possible. Lower lane counts can be configured. For example, the MAX9295D can be configured as dual 2-lane input port (i.e., 2x2 mode) by setting phy_config = 0b110 (2x4 mode), ctrl0_num_lanes = 0b01 (two data lanes), and ctrl1_num_lanes = 0b01 (two data lanes).

43.3.1.1 CSI-2 Serializer Port Architecture

The MAX9295A operates in 1(1x4) mode. PHY1 and PHY2 are combined to form a single four-lane port with the clock received on PHY2 (Figure 182). The MAX9295B can be configured via phy_config for either 1(1x4) or 2(1x2) mode. In 1x4 mode, PHY1 and PHY2 are combined to form a single four-lane port with the clock received on PHY2 (Figure 183). Note that this configuration is functionally equivalent to the MAX9295A in 1x4 mode. In 2x2 mode, PHY1 and PHY2 form independent ports and each PHY receives a clock (Figure 184). The MAX9295D and MAX9295E devices each have four PHYs available for use. Figure 185 below shows the architecture for 2(1x4) mode.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL Customers Under NDA

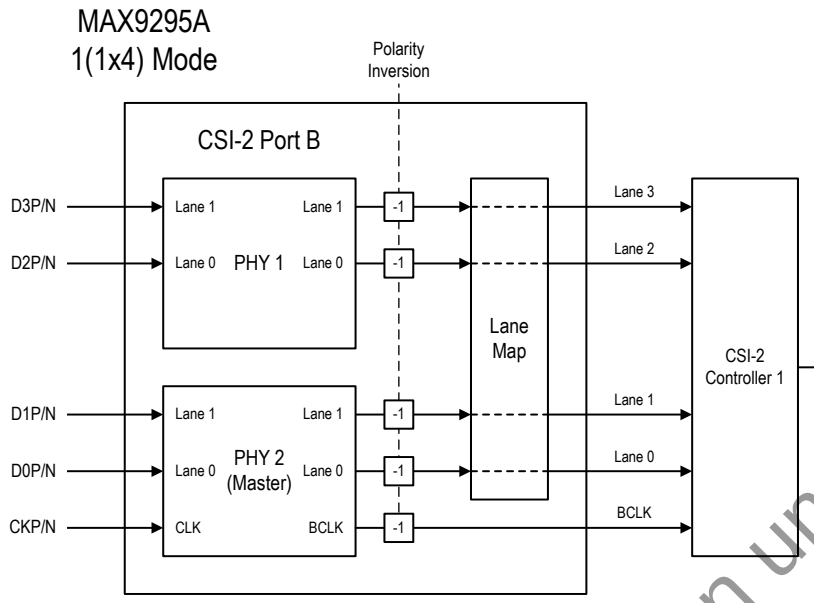


Figure 182. MAX9295A Serializer D-PHY Rx – 1(1x4) Mode

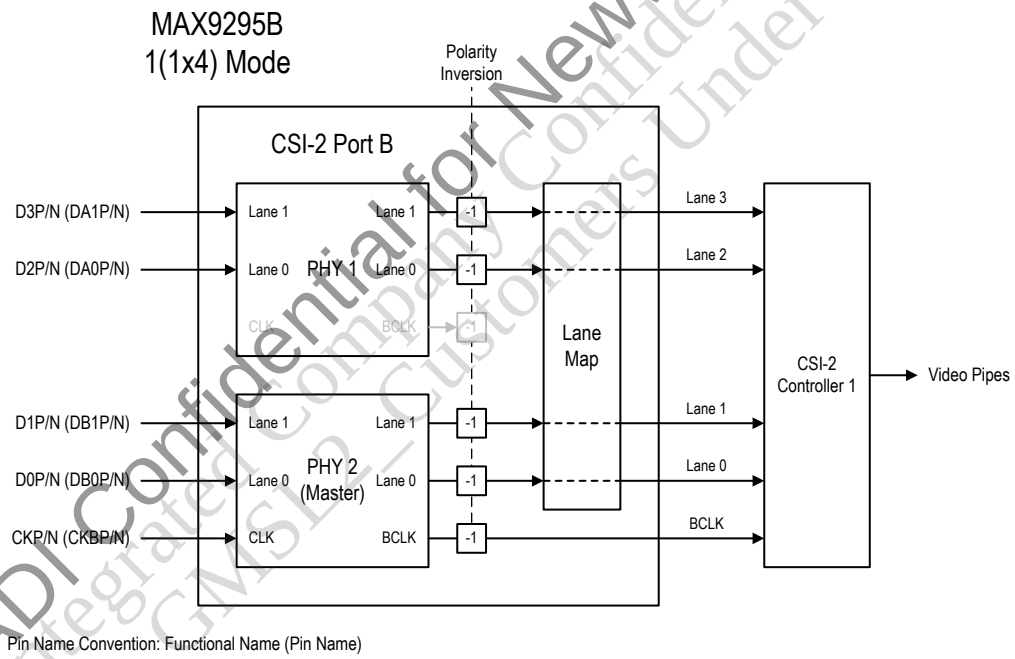


Figure 183. MAX9295B Serializer D-PHY Rx – 1(1x4) Mode

ADI Confidential for Newvision under NDA
Analog Devices, Inc.
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

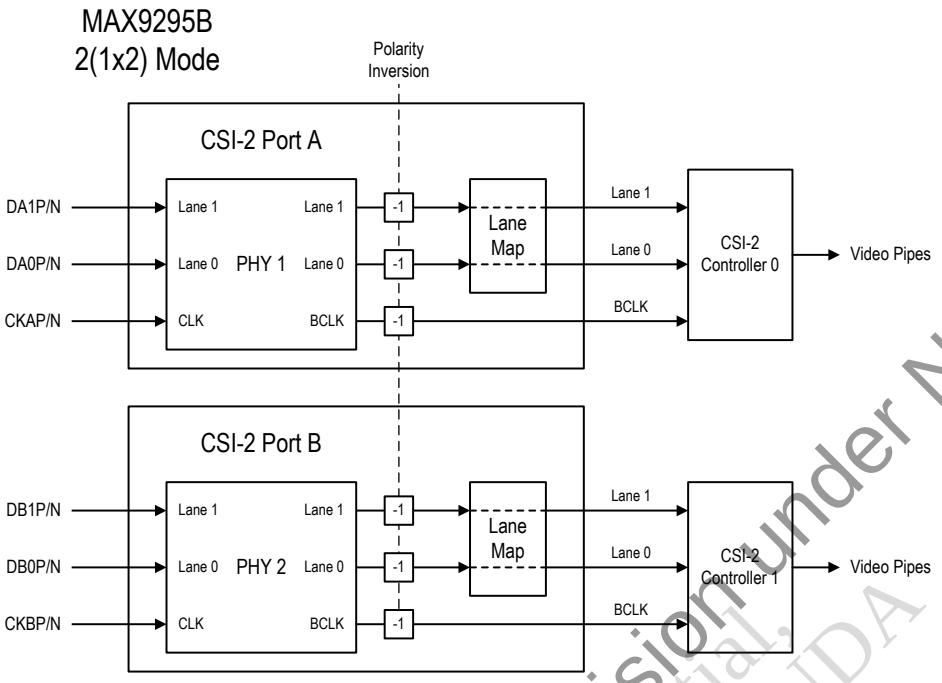


Figure 184. MAX9295B Serializer D-PHY Rx – 2(1x2) Mode

ADI Confidential for New vision under NDA
Analog Devices Confidential, Disclosure to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

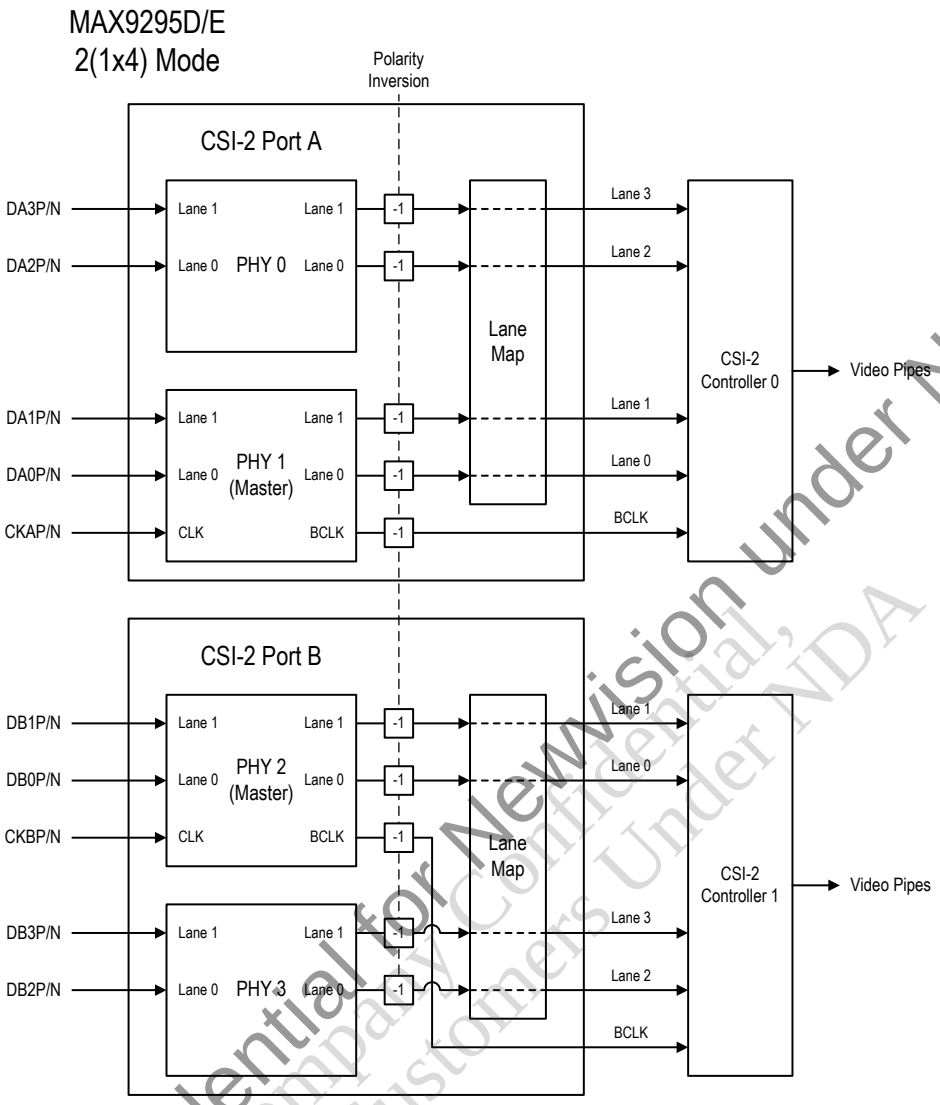


Figure 185. MAX9295D/E Serializer D-PHY Rx – 2(1x4) Mode

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

43.3.2 Serializer Lane Map/Swap and Polarity Inversion

To improve PCB layout/routing, each MIPI port supports lane reassignment of active lanes within the port. For example, D0 of Port A can be reassigned to D2 so that the D0 pins of the serializer can be connected to the D2 pins of the MIPI source. Clock lanes are fixed and cannot be re-assigned. The lane-mapping registers are subdivided by MIPI PHY, reference the figures in the CSI-2 Serializer Port Architecture section to determine the required lane mapping for a specific device and application.

Note: Multiple PHYs cannot connect to the same MIPI port lane. Valid lane map selection is dependent on the phy_config configuration. Refer to the appropriate *CSI-2 Serializer Port Architecture* block diagram.

Each data and clock lane supports polarity inversion. This allows a negative pin of the serializer to connect to a positive pin of the MIPI source (and vice-versa). The GMSL2 CSI-2 serializers' MIPI pinouts have a sequence of P, N, P, N, etc. when viewed counter-clockwise. If the MIPI source output has the opposite sequence, polarity inversion allows the MIPI output to connect directly to the PCB without routing through vias.

The lane map and polarity inversion registers are presented in Table 205. Lane swap is illustrated in the Lane Swap Example below.

Note: Lanes must be normally active before lane swapping can occur. For example, if the MAX9295A is programmed for two-lane mode, only D0 and D1 are active and can be used (D2 and D3 are not active). If the device is configured for four-lane mode, then all for lanes are active and available for lane swapping.

Table 205. GMSL2 CSI-2 Serializer Lane Map and Polarity Inversion Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x332	phy0_lane_map	3:0	0b1110	0bXX00: PHY0 lane 0 is MIPI Controller lane 0 0bXX01: PHY0 lane 0 is MIPI Controller lane 1 0bXX10: PHY0 lane 0 is MIPI Controller lane 2 0bXX11: PHY0 lane 0 is MIPI Controller lane 3 0b00XX: PHY0 lane 1 is MIPI Controller lane 0 0b01XX: PHY0 lane 1 is MIPI Controller lane 1 0b10XX: PHY0 lane 1 is MIPI Controller lane 2 0b11XX: PHY0 lane 1 is MIPI Controller lane 3
0x332	phy1_lane_map	7:4	MAX9295A: 0b1110 MAX9295B/D/E: 0b0100	0bXX00: PHY1 lane 0 is MIPI Controller lane 0 0bXX01: PHY1 lane 0 is MIPI Controller lane 1 0bXX10: PHY1 lane 0 is MIPI Controller lane 2 0bXX11: PHY1 lane 0 is MIPI Controller lane 3 0b00XX: PHY1 lane 1 is MIPI Controller lane 0 0b01XX: PHY1 lane 1 is MIPI Controller lane 1 0b10XX: PHY1 lane 1 is MIPI Controller lane 2 0b11XX: PHY1 lane 1 is MIPI Controller lane 3
0x333	phy2_lane_map	3:0	0b0100	0bXX00: PHY2 lane 0 is MIPI Controller lane 0 0bXX01: PHY2 lane 0 is MIPI Controller lane 1 0bXX10: PHY2 lane 0 is MIPI Controller lane 2 0bXX11: PHY2 lane 0 is MIPI Controller lane 3 0b00XX: PHY2 lane 1 is MIPI Controller lane 0 0b01XX: PHY2 lane 1 is MIPI Controller lane 1 0b10XX: PHY2 lane 1 is MIPI Controller lane 2

ADI Confidential for Review under NDA
Analog Devices Inc. Distributed only to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

				0b11XX: PHY2 lane 1 is MIPI Controller lane 3
0x333	phy3_lane_map	7:4	0b1110	0bXX00: PHY3 lane 0 is MIPI Controller lane 0 0bXX01: PHY3 lane 0 is MIPI Controller lane 1 0bXX10: PHY3 lane 0 is MIPI Controller lane 2 0bXX11: PHY3 lane 0 is MIPI Controller lane 3 0b00XX: PHY3 lane 1 is MIPI Controller lane 0 0b01XX: PHY3 lane 1 is MIPI Controller lane 1 0b10XX: PHY3 lane 1 is MIPI Controller lane 2 0b11XX: PHY3 lane 1 is MIPI Controller lane 3
0x334	phy0_pol_map	2:0	0b000	0bXX0: Normal Polarity for data lane 0 0bXX1: Inverse Polarity for data lane 0 0bX0X: Normal Polarity for data lane 1 0bX1X: Inverse Polarity for data lane 1 0b0XX: Normal Polarity for clock lane 0b1XX: Inverse Polarity for clock lane
0x334	phy1_pol_map	6:4	0b000	0bXX0: Normal Polarity for data lane 0 0bXX1: Inverse Polarity for data lane 0 0bX0X: Normal Polarity for data lane 1 0bX1X: Inverse Polarity for data lane 1 0b0XX: Normal Polarity for clock lane 0b1XX: Inverse Polarity for clock lane
0x335	phy2_pol_map	2:0	0b000	0bXX0: Normal Polarity for data lane 0 0bXX1: Inverse Polarity for data lane 0 0bX0X: Normal Polarity for data lane 1 0bX1X: Inverse Polarity for data lane 1 0b0XX: Normal Polarity for clock lane 0b1XX: Inverse Polarity for clock lane
0x335	phy3_pol_map	6:4	0b000	0bXX0: Normal Polarity for data lane 0 0bXX1: Inverse Polarity for data lane 0 0bX0X: Normal Polarity for data lane 1 0bX1X: Inverse Polarity for data lane 1 0b0XX: Normal Polarity for clock lane 0b1XX: Inverse Polarity for clock lane

ADI Confidential for Newvision Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

43.3.2.1 Lane Swap Example

Lane swap is illustrated in Figure 186 with a MAX9295A device. The register writes required for the depicted lane mapping are presented in Table 206 (reference the bolded decode entries).

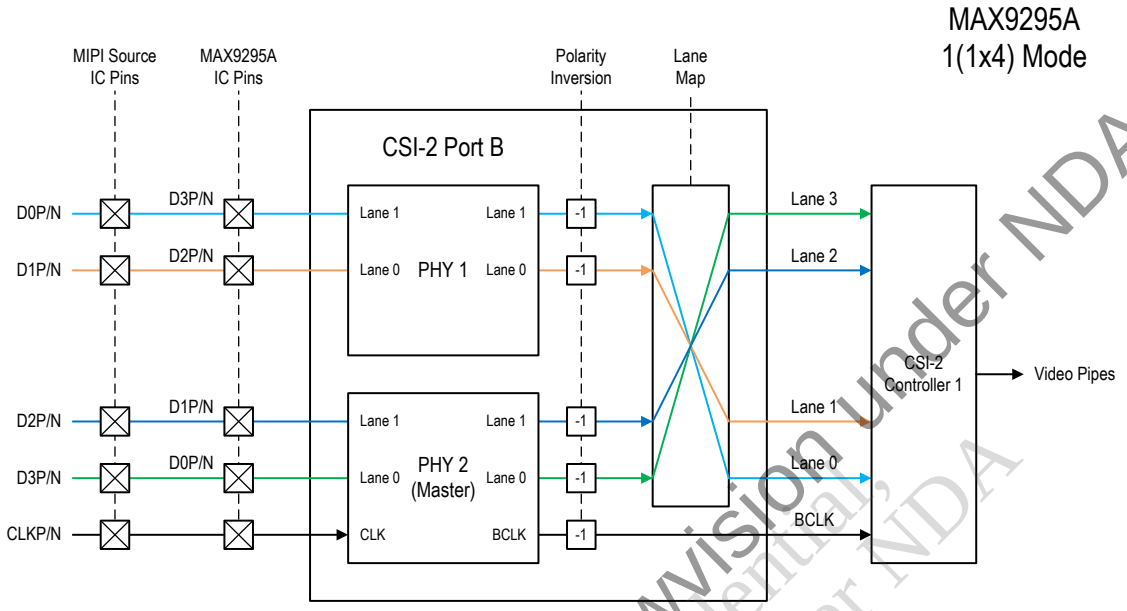


Figure 186. MAX9295A Serializer – Lane Swap

Table 206. MAX9295A Serializer Lane Swap Example Register Writes

Register Address	Bitfield Name	Bits	POR	Decode
0x332	phy1_lane_map	7:4	MAX9295A: 0b1110 MAX9295B/D/E: 0b0100	0bXX00: PHY1 lane 0 is MIPI Controller lane 0 **0bXX01: PHY1 lane 0 is MIPI Controller lane 1** 0bXX10: PHY1 lane 0 is MIPI Controller lane 2 0bXX11: PHY1 lane 0 is MIPI Controller lane 3 **0b00XX: PHY1 lane 1 is MIPI Controller lane 0** 0b01XX: PHY1 lane 1 is MIPI Controller lane 1 0b10XX: PHY1 lane 1 is MIPI Controller lane 2 0b11XX: PHY1 lane 1 is MIPI Controller lane 3
0x333	phy2_lane_map	3:0	0b0100	0bXX00: PHY2 lane 0 is MIPI Controller lane 0 0bXX01: PHY2 lane 0 is MIPI Controller lane 1 0bXX10: PHY2 lane 0 is MIPI Controller lane 2 **0bXX11: PHY2 lane 0 is MIPI Controller lane 3** 0b00XX: PHY2 lane 1 is MIPI Controller lane 0 0b01XX: PHY2 lane 1 is MIPI Controller lane 1 **0b10XX: PHY2 lane 1 is MIPI Controller lane 2** 0b11XX: PHY2 lane 1 is MIPI Controller lane 3

ADI Confidential for Newvision under NDA
Analog Devices Initial
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.3.2.1.1 Lane Swap Example Programming Summary

The following is a summary of the programming used in Figure 186:

- 0x332 = 0x10
  - phy0_lane_map = b0000 (Note: PHY0 is not available on the MAX9295A, use the default value.)
  - phy1_lane_map = b0001
- 0x333 = 0x0B
  - phy2_lane_map = b1011
  - phy3_lane_map = b0000 (Note: PHY3 is not available on the MAX9295A, use the default value.)

43.3.3 D-PHY Deskew

The MIPI D-PHY requires deskew calibration when operating above 1.5Gbps/lane. Each data lane supports clock-to-data lane deskew calibration. However, deskew is not programmed by the CSI-2 Configuration tool (GUI). See the

MIPI D-PHY Deskew section for configuration details.

ADI Confidential for NewVision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.3.4 Video Pipes and GMSL Core

Video pipes allow GMSL2 systems to serve as a bridge between different input and output video interfaces. Within a GMSL2 system, the video data between the input and output interfaces is converted into a generic parallel video data format consisting of pixel data, PCLK, HS, VS, DE, and any other interface-specific control signals.

43.3.4.1 Controller-to-Pipe Mapping

Video data is mapped from the two MIPI controllers to the video pipes using the FRONTTOP address space. A pipe can only be connected to a single controller, but a controller can connect to any combination of video pipes. The default routing for CSI-2 serializers routes data from Controller 0 to pipe X and from Controller 1 to pipe Z (Figure 187). The routing from controller to video pipe is configurable; the routing from port to controller is fixed. Controller-to-pipe mapping registers are presented in Table 207.

Note: Port-to-controller mapping is fixed; controller-to-video pipe mapping is configurable via register programming. The FRONTTOP address space refers to routing from port to video pipe. Since the port-to-controller mapping is fixed, this is the effective routing that is configured.

Note: Not all features are available for GMSL1 data. Reference the GMSL1 Legacy Mode (Backward Compatibility) section for more information.

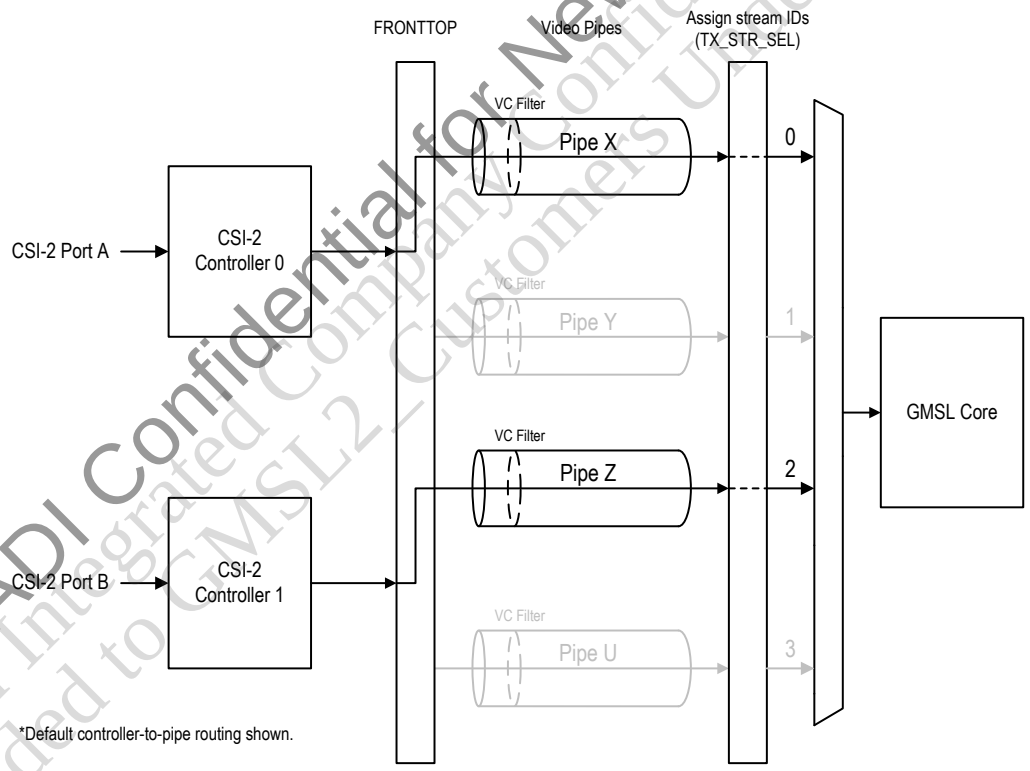


Figure 187. GMSL2 CSI-2 Serializer Controller-to-Pipe Routing

ADI Confidential for Newvision under NDA
Analog Devices, Inc. Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Controller-to-pipe routing requires the following configuration:

- Enable the port with the START_PORT{A,B} bitfield.
- Configure controller-to-pipe routing with the CLK_SEL{U,Z,Y,X} and START_PORT{A,B}{U,Z,Y,X} bitfields.
    - When programming START_PORT{A,B}{U,Z,Y,X}, each video pipe can only connect to a single controller. A controller can connect to more than one video pipe.
    - When programming a two-port device, ensure that the START_PORT{A,B}{U,Z,Y,X} settings for the different ports do not connect to the same video pipe.

Table 207. GMSL2 CSI-2 Serializer Controller-to-Pipe Mapping Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x308	CLK_SEL{U,Z,Y,X}	3:0	0b1100	0b0: Map Port A to the associated pipe 0b1: Map Port B to the associated pipe
0x308	START_PORT{B,A}	5:4	0b11	Enable the associated CSI port
0x311	START_PORTB{U,Z,Y,X}	3:0	b0101	Start Video Pipe from CSI Port B. Ensure START_PORTA[3:0] & START_PORTB[3:0] = 0 when using a two port device.
0x311	START_PORTA{U,Z,Y,X}	7:4	b0101	Start Video Pipe from CSI Port A. Ensure START_PORTA[3:0] & START_PORTB[3:0] = 0 when using a two port device.
0x314–0x31B	mem_dt{1,2}_sel{x,y,z,u}	6:0	0b0000000	Pixel data type routing. Set the 6 LSBs to designate the DT that will be routed to the corresponding video pipe. Set the MSB (bit 6) to enable.
0x3D8–0x3DF	mem_dt{7,8}_sel{x,y,z,u}	6:0	0b0000000	Pixel data type routing. Set the 6 LSBs to designate the DT that will be routed to the corresponding video pipe. Set the MSB (bit 6) to enable.
0x3C0–0x3CF	mem_dt{3-4}_sel{x,y,z,u}	7:0	0x00	Select a designated data type to route to Video Pipeline for VS.
0x3C0–0x3CF	mem_dt{5-6}_sel{x,y,z,u}	7:0	0x00	Select a designated data type to route to Video Pipeline for HS.
0x3D0–0x3D1	mem_dt{3-6}_sel{x,y,z,u}_en	7:0	0x00	Enable DT routing for sync signals.
0x309–0x310	VC_SEL{X,Y,Z,U}_{L,H}	15:0	0xFFFF	Sets which VC can be routed to the associated video pipe. Each bit position correlates to the corresponding VC (bit 0 = VC0, bit 1 = VC1, etc.). Only VC0-3 are supported. Reset value allows any VC in any pipe.
0x314	VS_VC_CHECK	7	0b0	0b0: Datatype or VC check disabled for VS 0b1: Datatype or VC check enabled for VS
0x315	independent_vs_mode		0b0	0b0: independent VS disabled 0b1: independent VS enabled

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA

43.3.4.2 Selecting Pipe Routing

Controller-to-pipe routing is dependent on system design and requirements. Factors that determine the minimum number of pipes include the number of input ports, CSI interleaving (VS), datatypes (and bpp), and output ports (including data aggregation, duplication, and splitting requirements).

Each CSI-2 input port requires a minimum of one video pipe. For a device configured for two ports, this means that at least two video pipes will be used. Each video pipe has a single FIFO buffer which connects to a single controller source. However, one controller can connect to any number of video pipes. Aggregation is performed on the deserializer.

A virtual channel is an independent video stream with its own timing signals. Video pipes typically transmit a single VS signal. If a CSI-2 input consists of multiple interleaved VCs, the interleaving pattern must be known so that the VS signals can be properly routed through the video pipes. Reference the Short-Packet (VS) Processing for additional information.

All data within a video pipe must share bpp/PCLK values, and video pipes have a maximum bpp rate of 24. When multiple datatypes and/or bpp rates are received on a CSI-2 controller, they can be routed to the video pipes using the following methods:

1. Maintain constant bpp rate in a pipe by only routing input data with the same bpp.
    a. **Example**: RAW8 and EMB8 can share a pipe, but RAW12 is routed to a separate pipe.

2. Maintain constant bpp rate in a pipe by manipulating input data using double and UDT (triple) modes (see the Double Mode section for details).
    a. Triple loaded 8-bit datatypes can only share a pipe with other triple loaded 8-bit datatypes and 24bpp datatypes. It cannot be used with double loaded or zero padded data.
    b. **Example**: EMB8 (DBL) and RAW16 share a pipe, provided EMB8 is doubled in the serializer (two input pixels are combined into a single GMSL pixel).

3. Maintain constant bpp rate in a pipe by zero padding input data.
    a. This requires that 8 ≤ bpp ≤ 16 for all bpp rates within the pipe.
        i. Zero padding occurs after doubling. The zero-padding bpp requirement applies to the resulting bpp after doubling.
    b. This method reduces the GMSL bandwidth efficiency (i.e., less incoming video bandwidth can be supported at a given GMSL2 link rate). However, it can make device configuration simpler and reduce power consumption by disabling unused pipes.
    c. PCLK drift detection must be disabled when using this feature.
    d. **Example**: EMB8 (DBL), RAW12, and RAW16 share pipe X. EMB8 is doubled to 16 bpp. RAW12 is zero padded to 16bpp. RAW16 is unmodified. All datatypes will be 16bpp inside of the pipe.

Within the above listed constraints, any combination of input bpp manipulations can be used simultaneously to meet the requirement of constant bpp/PCLK within a video pipe (native input bpp, double mode, UDT (triple), and zero padding).

Analog Devices Confidential for New Version under NDA
Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA

Incoming video data is processed as the equivalent bpp rate by decoding the datatype hex code in the CSI-2 header (Table 208).

Pipe X is unique and offers extended features not available on other video pipes:

- The watermark safety feature is only available for 24-bit RGB888 video data on pipe X.
- Parallel input data is automatically routed to pipe X.

43.3.4.3 Pixel Datatype and Virtual Channel Routing

Every CSI-2 packet includes a header that indicates the datatype (DT) and virtual channel (VC). This information can be used to route the incoming data throughout the serial link system.

The data within a controller can be split and routed to different video pipes by using DT and/or VC filtering. When no DT or VC filters are enabled for a pipe (default), all data received on a controller will be routed to the pipes per the Controller-to-Pipe Mapping set by CLK_SEL{U,Z,Y,X} and START_PORT{A,B}{U,Z,Y,X}.

The mem_dt_sel registers (Table 207) are used to filter the controller data by CSI-2 datatype code (Table 208). When using datatype filtering, up to four datatypes can be routed from the controller to the selected pipe. The pixel datatype codes to be routed must be set in mem_dt {1,2,7,8}_sel{x,y,z,u} for the desired pipe. The 6 LSBs in the register must match the incoming datatype code. Setting bit 7 enables the filter.

In pixel mode, when multiple data streams are transmitted over the same pipe with different virtual channels, the VS_VC_L and VS_VC_H bits must be set to represent the virtual channels present on that pipe. Each bit place represents a virtual channel within these registers. For example; if VS_VC0_L[0] = 1 and VS_VC0_L[1] = 1, then pipe 0 would expect to have virtual channels 0 and 1 on the pipe.

Table 208. Typical CSI-2 Data Type Hex Codes

CSI-2 Long Packet	Data ID	Bits Per Pixel (bpp)	Notes
RAW8	0x2A	8	
RAW10	0x2B	10	
RAW12	0x2C	12	
RAW14	0x2D	14	
RAW16	0x2E	16	
RAW20	0x2F	20	
YUV422-8	0x1E	16	
YUV422-10	0x1F	20	
RGB565	0x22	16	
RGB666	0x23	18	
RGB888	0x24	24	
Embedded	0x12	8	
Null	0x10	8	
Blank	0x11	8	
User-Defined	0x31-0x37	8	
YUV422-12	0x30	24	Not CSI-2 standard (custom data type)

VC filtering is configured with the VC_SEL bitfield for each video pipe. Each bit in the VC_SEL bitfield corresponds to a virtual channel equal to the bit position. For example, bit 0 of VC_SEL corresponds to VC0, and bit 3 of VC_SEL corresponds to VC3. When a bit position is set to zero, that VC will be not be allowed to enter that pipe.

ADI Confidential for new vision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

All VC_SEL bits POR to b1 (allowing all VCs). The VC information is transmitted to the deserializer, allowing further VC routing on the output.

Note: MAX9295A/B/D/E devices only support VC0-3.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.3.4.3.1 Serializer Video Pipe Routing Example

Figure 188 shows video pipe routing in an example GMSL2 CSI-2 system. The input to Controller 1 consists of two datatypes on the same VC. Since the bpp rates of the two datatypes are different, the datatypes are routed to two different video pipes. The deserializer stream IDs are used to duplicate the data in the deserializer, allowing duplicate output on two separate MIPI ports (see the Data Replication section for details). The routing settings used are shown below.

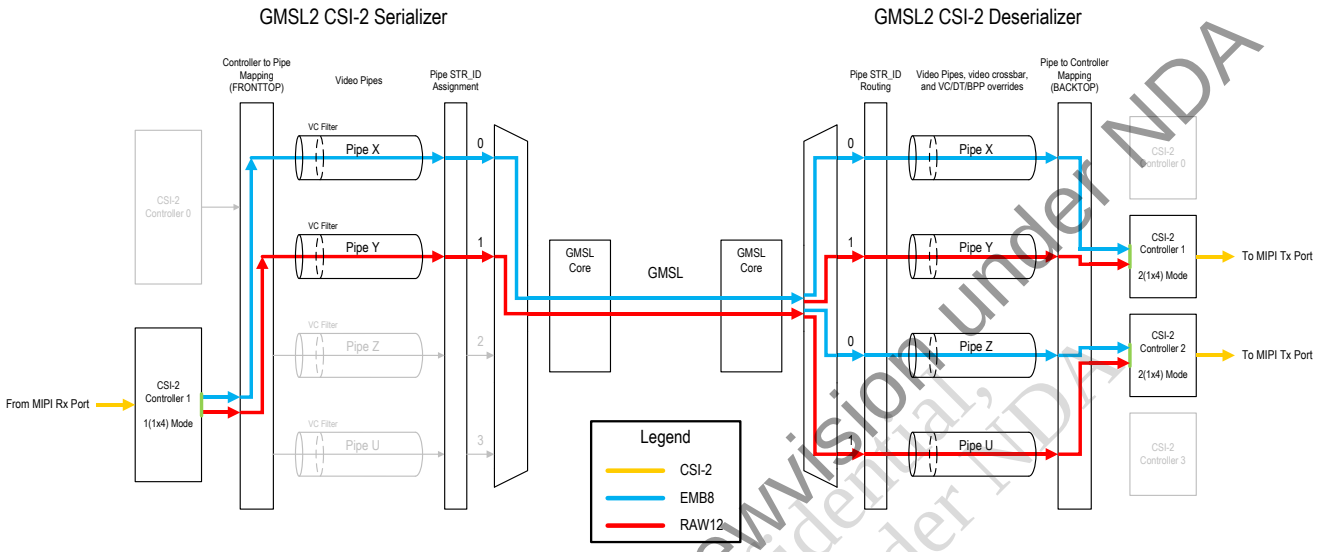


Figure 188. GMSL2 CSI-2 Video Pipes – System Routing

EMB8 and RAW12 Example:

Serializer Routing Settings (*Figure 188*):

- START_PORTA{U,Z,Y,X} = 0b0000 – Port A is unused.
- START_PORTB{U,Z,Y,X} = 0b0011 – Enable pipe X and Y from port B.
- CLK_SELX = 0b1 – Enable pipe X from port B.
- CLK_SELY = 0b1 – Enable pipe Y from port B.
- START_PORTB = 0b1 – Enable port B.
- VID_TX_EN_X = 0b1 – Enable video Tx from pipe X.
- VID_TX_EN_Y = 0b1 – Enable video Tx from pipe Y.
- mem_dt1_selx = 0x52 – Route only EMB8 (DT 0x12) to pipe X.
- mem_dt1_sely = 0x6C – Route only RAW12 (DT 0x2C) to pipe Y.

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

43.3.4.4 Short-Packet (VS) Processing

CSI-2 short packets are mapped to parallel video and VS (FS/FE) signals (see the CSI-2 Packets section). Each video pipe contains a single set of SYNC signals. These are routed from the controller to the pipes in one of two ways:

1. Combined Vertical Sync Mode (Default and recommended)

A VS signal is generated from the FS/FE packets for every VC on the controller. All VS signals in the controller are then ORed together. The resulting signal is used as the VS for every pipe connected to that controller. An FS packet creates a rising edge on VS; an FE packet creates a falling edge on VS.

The VS_VC_CHECK bit determines whether short packets (FS/FE) are filtered using the mem_dt{3-6}_sel{x,y,z,u} registers. By default, this is disabled (VS_VC_CHECK = b0) and only pixel data types are filtered. The VS of a pipe is determined by the process described above. Setting VS_VC_CHECK = b1 means FS/FE (DT 0x00 and DT 0x01, respectively) must be routed to a pipe using the mem_dt_sel registers.

2. Independent Vertical Sync Mode

When independent_vs_mode = b1, a video pipe VS is generated by only the FS/FE packets that are associated with the VCs routed to the pipe by the VC_SEL register. When independent_vs_mode = b1, datatype filtering for short packets using the mem_dt{3-6}_sel{x,y,z,u} registers is disabled. This mode should be used when the input has multiple VCs that are not frame-synchronized. Each VC must be routed to a dedicated pipe.

ADI Confidential for New Version under NDA
Analog Devices Confidential - Limited to only those with a need to know
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

43.3.4.4.1 VS Routing

The CSI-2 protocol allows flexibility in the structure of data transmissions. In GMSL2 systems, the VS signal(s) and datatype bpp rate(s) of the input signal must be considered when configuring video pipe routing. The use cases presented in this section evaluate CSI-2 input signals in terms of the VS signal (Figure 189). Note that the use cases presented here are not exhaustive and should be used as a guide when configuring a system for a given application.

GMSL2 systems have a limited number of video pipes. Although each virtual channel can be routed to separate video pipes, extended applications may require that virtual channels share video pipes. Virtual channels may share a video pipe only if the bpp rates align and the VS signals meet the relevant criteria discussed below.

43.3.4.4.1.1 VS Routing Use Cases

- **Single Virtual Channel**
  - Case 1.a) Single Virtual Channel, Single Datatype
  - Case 1.b) Single Virtual Channel, Multiple Datatypes (Embedded Data)
    - If a virtual channel comprising multiple datatypes is split into multiple pipes (due to different bpp rates), the VS will be duplicated on the serializer. Only one VS should be used when routing to the deserializer output. See Figure 190 and Figure 191.
- **Multiple Virtual Channel**
  - Case 2.a) Frame-Synchronized Line-Interleaved (Multiple Virtual Channels share a VS)
    - Since all virtual channels effectively have the same VS, they can always share a video pipe if the bpp rates align. Note that the deserializer programming is dependent on whether the virtual channels share a video pipe or are split into different video pipes. See Figure 192 and Figure 193.
  - Case 2.b) Unsynchronized Line-Interleaved (Multiple Virtual Channels with independent, possibly overlapping, VS)
    - Since the VS signals may be overlapping, each virtual channel must be routed to separate video pipes to maintain the independent VS signals. See Figure 194 and Figure 195.
  - Case 2.c) Frame-Interleaved (Multiple Virtual Channels with non-overlapping VS)
    - The VS signals are independent, but they are never high at the same time. The MAX9296 requires that each virtual channel be routed to a different video pipe. Other deserializers can support frame-interleaved virtual channels sharing a video pipe if the bpp rates align. See Figure 196 and Figure 197.

Note: The virtual channels shown in the Multiple Virtual Channel use cases can comprise any combination of cases 1.a and 1.b.

Deserializer configuration information is presented in *Short-Packet (VS) Processing* below.

ADI Confidential for Navision under NDA
Analog Devices Confidential Distribution Only under NDA
Maxim Integrated Company Confidential
Provided to GMSL Customers Under NDA

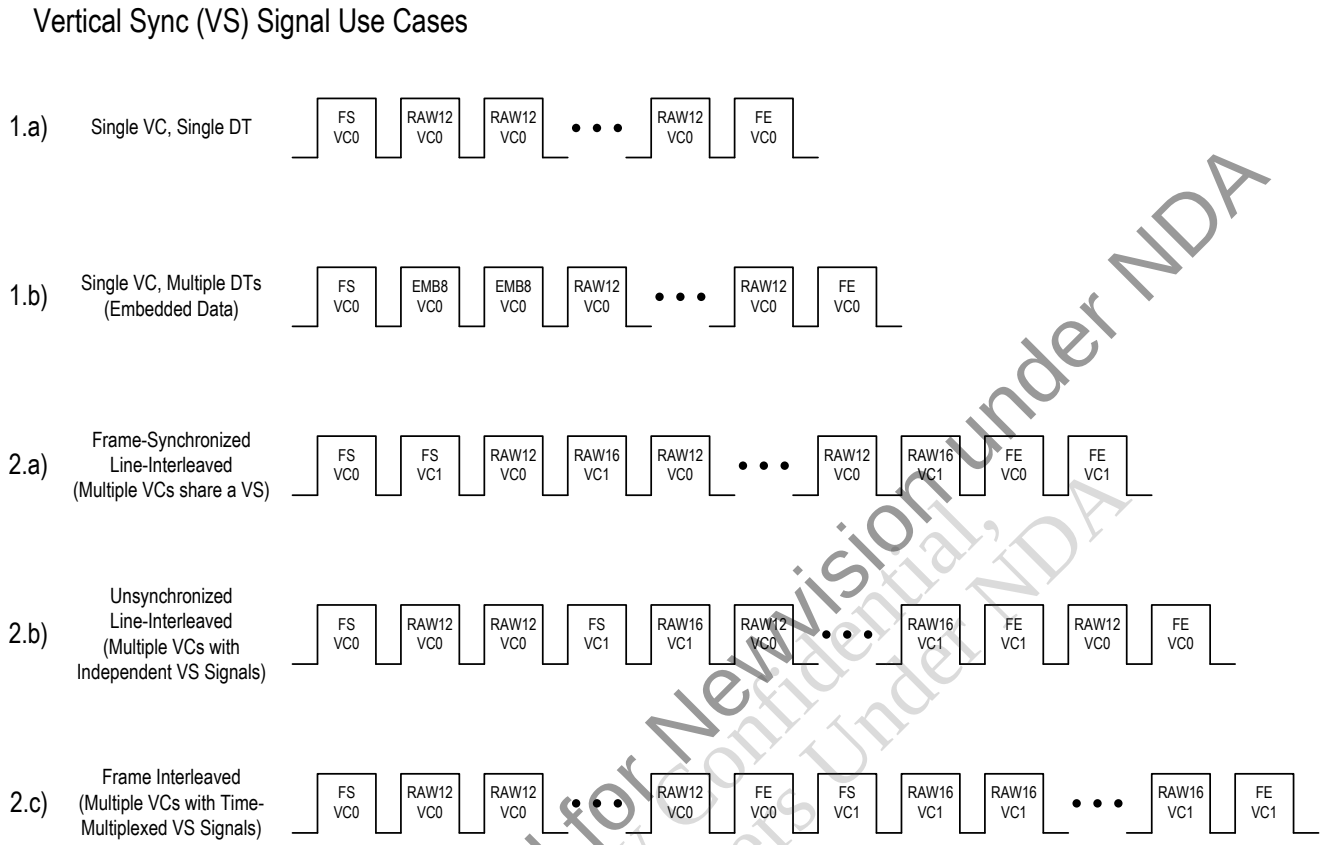


Figure 189. CSI-2 VS Routing Use Cases

ADI Confidential for NewVision under NDA
Analog Devices Confidential,
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Case 1.b) Serializer CTRL and Video Pipes

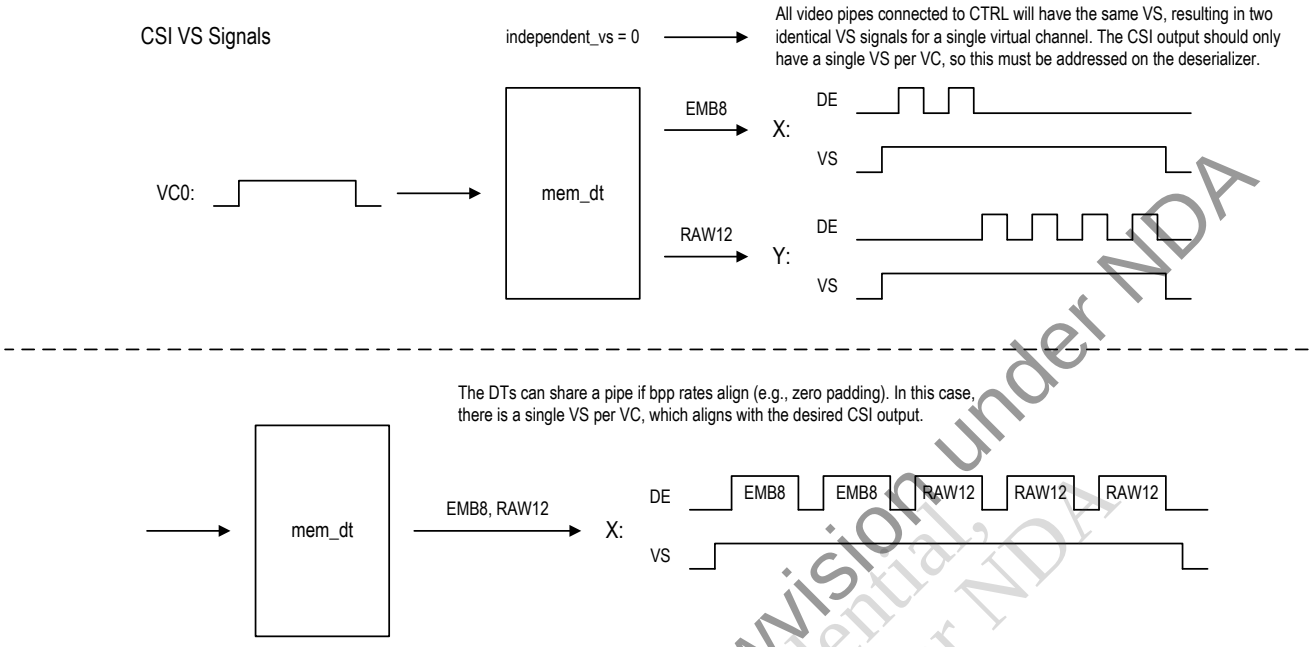


Figure 190. Case 1.b) Serializer – Embedded Data Input

Case 1.b) Deserializer

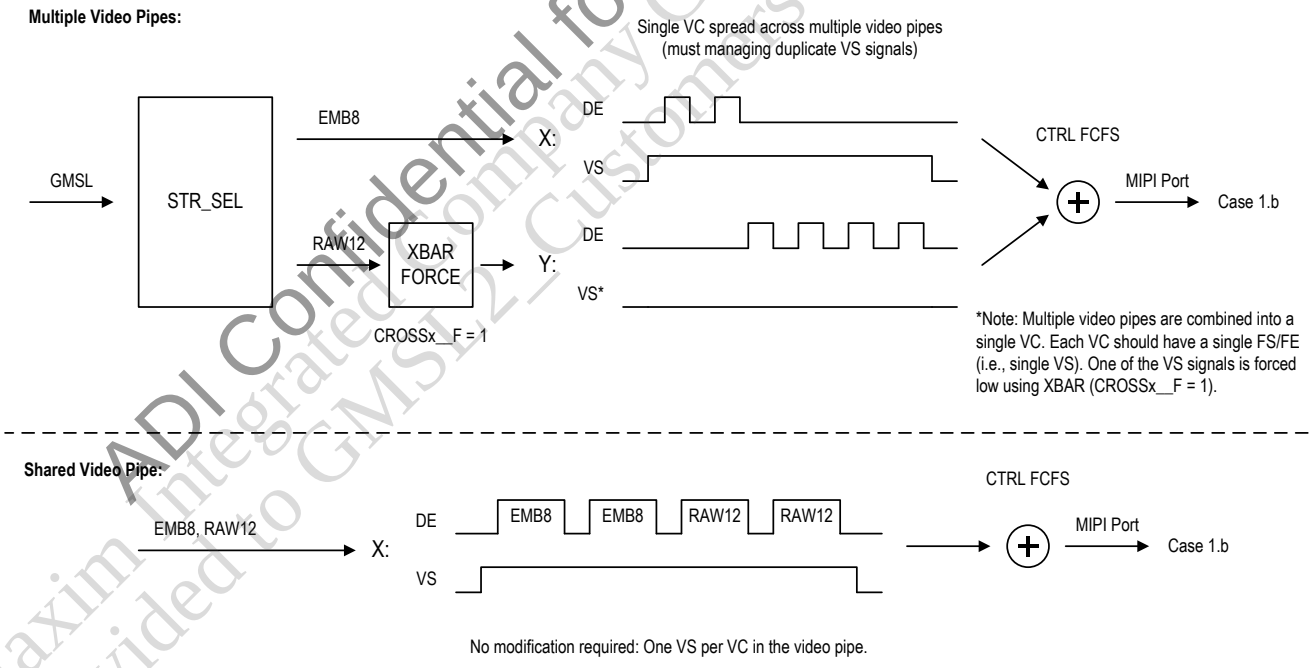


Figure 191. Case 1.b) Deserializer – Embedded Data Input

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Case 2.a) Serializer CTRL and Video Pipes

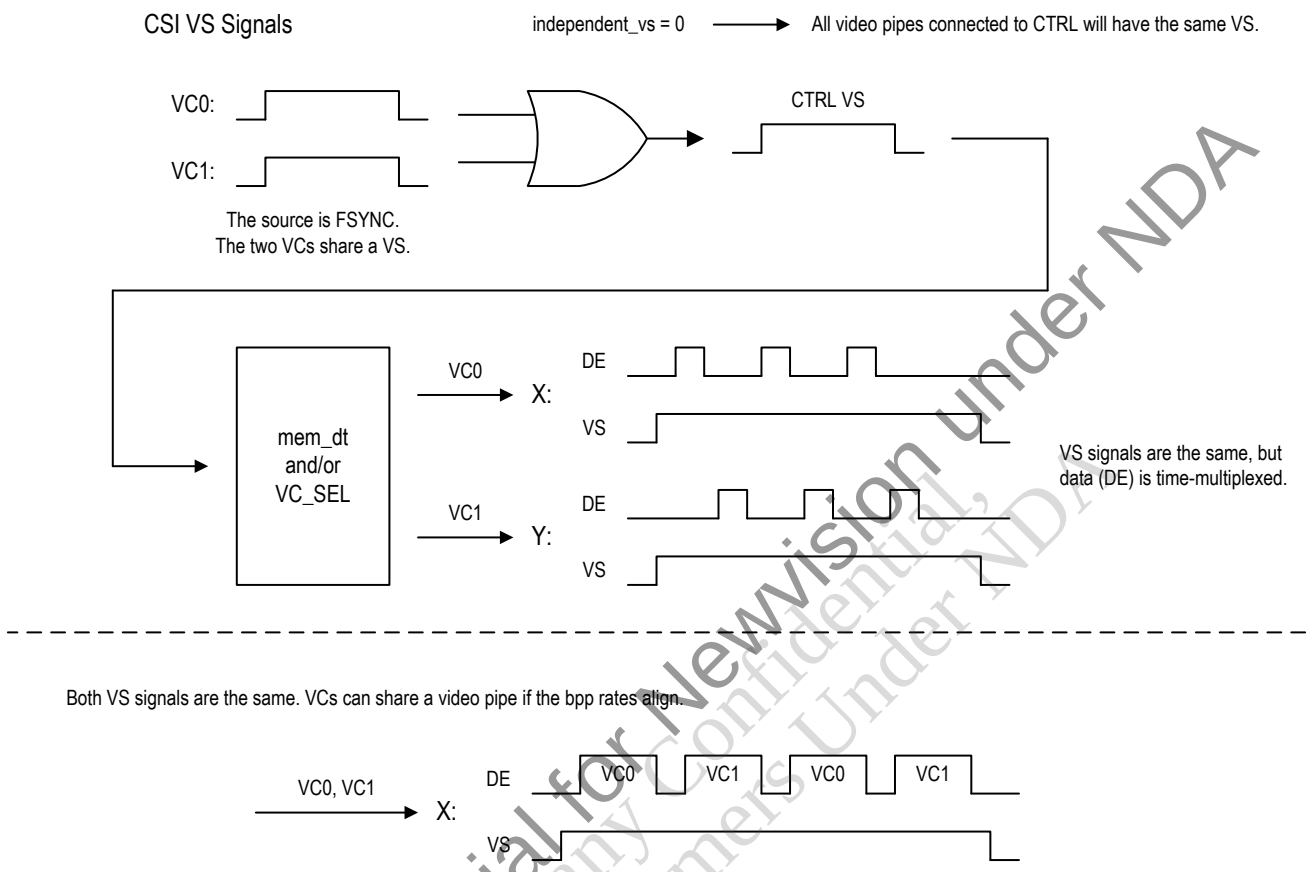


Figure 192. Case 2.a) Serializer – Frame-Synchronized Line-Interleaved Input

ADI Confidential for Newvision under NDA
Analog Devices, Inc.
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Case 2.a) Deserializer

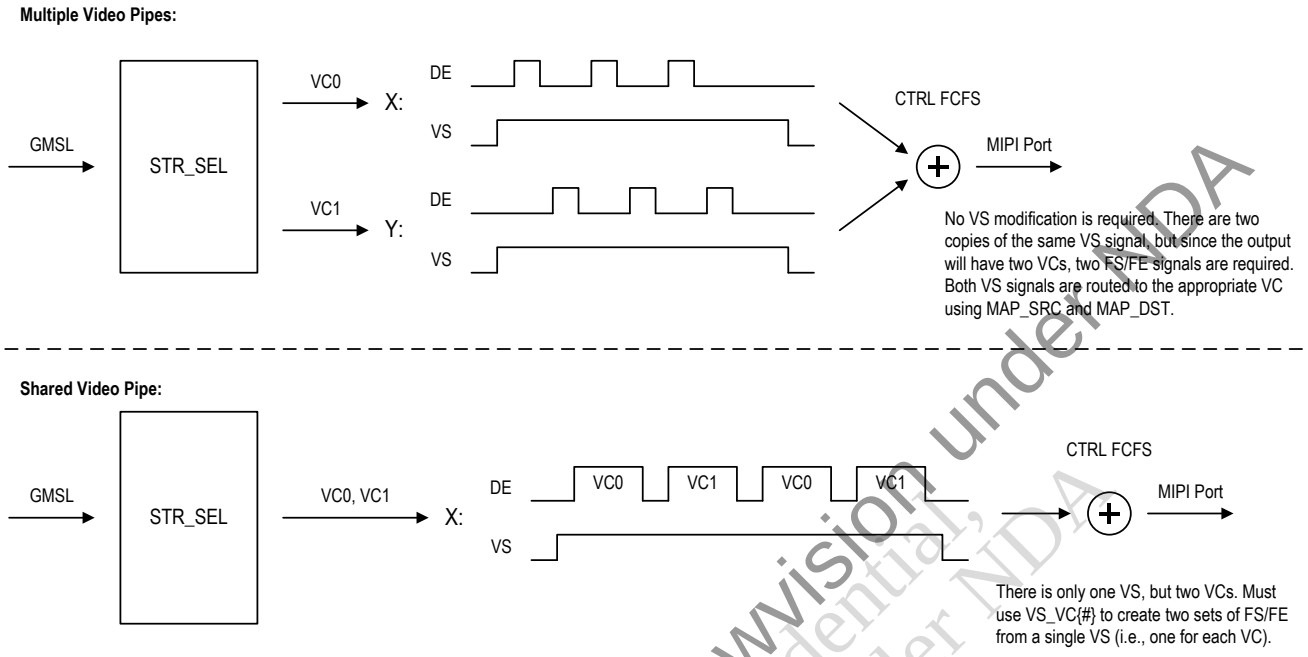


Figure 193. Case 2.a) Deserializer – Frame-Synchronized Line-Interleaved Input

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Case 2.b) Serializer CTRL and Video Pipes

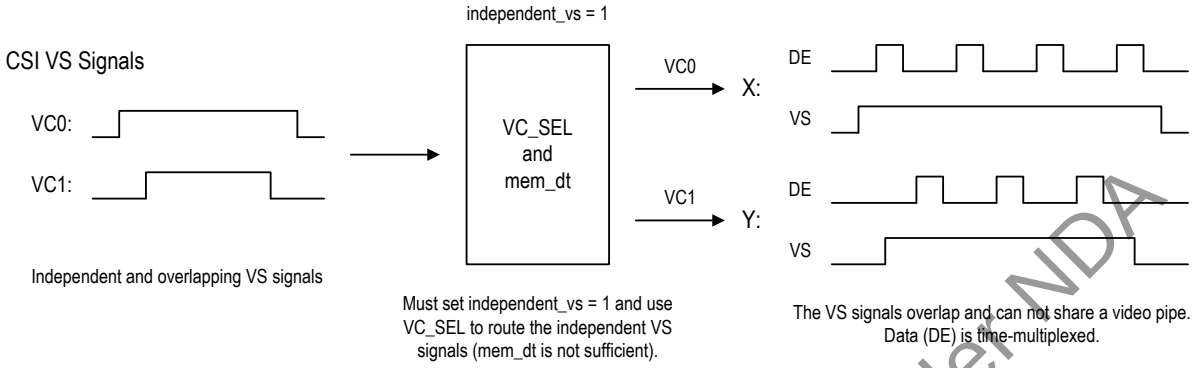


Figure 194. Case 2.b) Serializer – Unsynchronized Line-Interleaved Input

Case 2.b) Deserializer

There is only one VS per VC. Use FCFS routing to route all VS signals to the output (same method used in example 2.a multiple video pipes).

Figure 195. Case 2.b) Deserializer – Unsynchronized Line-Interleaved Input

ADI Confidential for New vision under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Case 2.c) Serializer CTRL and Video Pipes

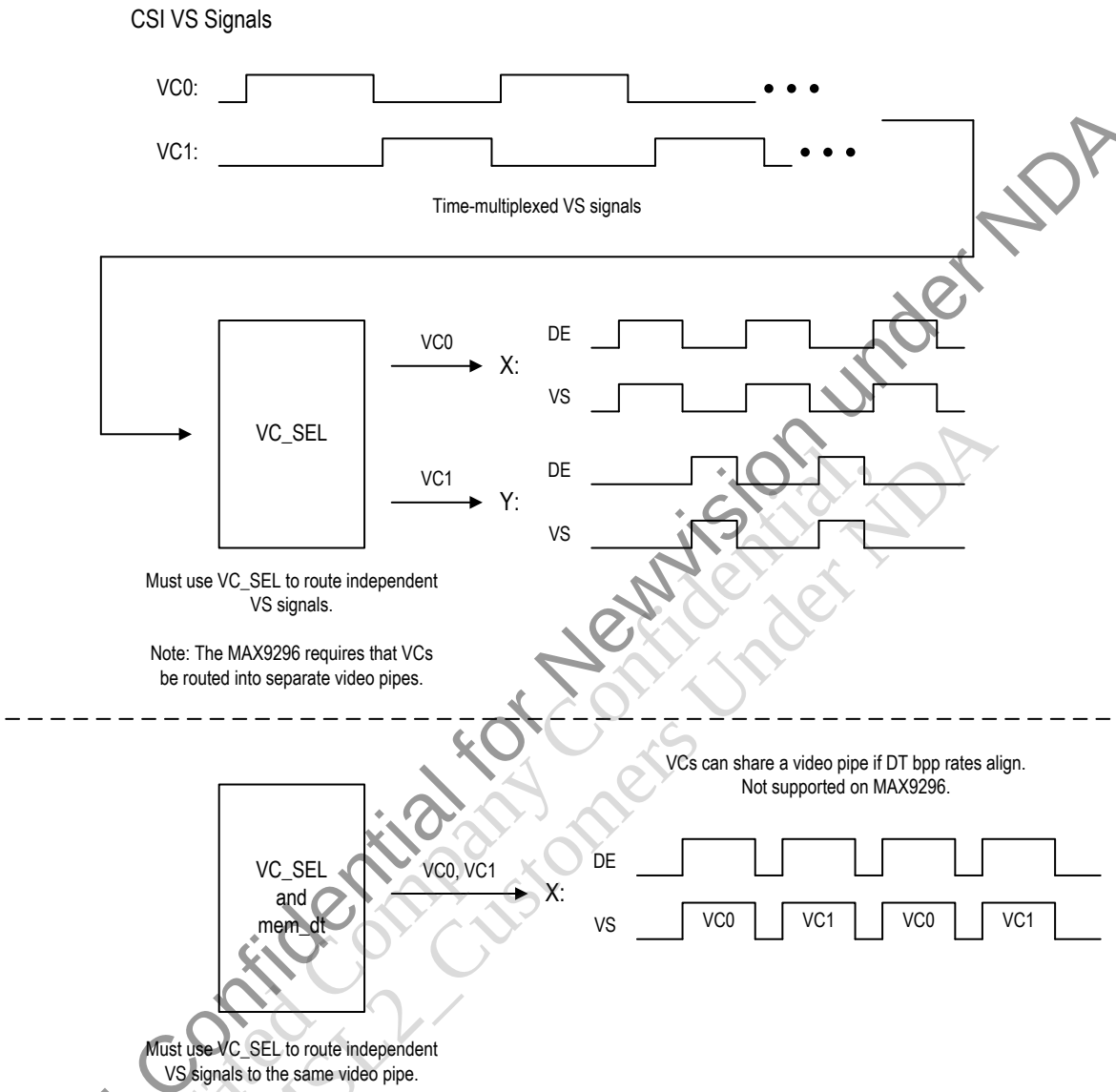


Figure 196. Case 2.c) Serializer – Frame-Interleaved Input

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

Case 2.c) Deserializer

Multiple Video Pipes:

Use VC FCFS routing (i.e., the method used in example 2.a multiple video pipes).

Shared Video Pipe:

When multiple VCs share a video pipe, add all VS FS/FE to FCFS routing for the video pipe.

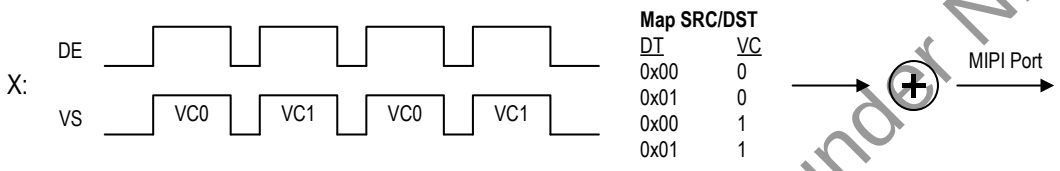


Figure 197. Case 2.c) Deserializer – Frame-Interleaved Input

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.3.4.5 Input BPP Manipulation

43.3.4.5.1 Double Mode (Serializer)

Double mode is a data arrangement available for datatypes with a bpp = 8, 10, and 12. With double mode enabled, two input pixels are concatenated and processed as a single pixel within a video pipe. This concatenation reduces the internal PCLK and increases the GMSL2 bandwidth efficiency. Double mode is enabled on a per pipe and per bpp basis. Programing details are in *Table 209*; refer to the Interface-Specific Bandwidth Calculations section for additional information and calculations.

Further, some 8-bit datatypes (i.e., EMB8 and DTs 0x31–0x37) can be combined and transmitted by the serializer as 24-bit (see Table 208). Set ctrl0_mode_UDT and/or ctrl1_mode_UDT = b1 to treat these datatypes as 24 bpp. This mode cannot be used simultaneously with 8 bpp double mode.

Note: When ctrl0_mode_UDT and/or ctrl1_mode_UDT = b1, the specified 8-bit datatypes in the configured controller are treated as 24-bit datatypes.

When using double or triple mode, the new internal bpp must be programed into the serializer in addition to enabling the mode. Each video pipe has a soft_bpp bitfield that must be set to the new bpp (e.g., 8 → 16, 8 → 24, 10 → 20, 12 → 24) and a soft_bpp_en bit.

Note: The connected GMSL2 deserializer must also be set appropriately to revert the concatenated pixels to the original size.

See the Double Mode Example Configurations section for examples of the writes required to properly configure double mode with a serializer and deserializer pair.

Note: The terms UDP and UDT are functionally equivalent and refer to the generic “User-Defined 8-bit Data Types” defined in MIPI CSI-2 v1.3.

Table 209. GMSL2 CSI-2 Serializer Double Mode Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x312	bpp8dbl{U,Z,Y,X}	3:0	0b0000	0b0: Send as 8-bit pixels 0b1: Send 8-bit pixels as 16-bit pixels
0x313	bpp10dbl{U,Z,Y,X}	3:0	0b0000	0b0: Send as 10-bit pixels 0b1: Send 10-bit pixels as 20-bit pixels
0x313	bpp12dbl{U,Z,Y,X}	7:4	0b0000	0b0: Send as 12-bit pixels 0b1: Send 12-bit pixels as 24-bit pixels
0x336	ctrl0_mode_UDT	5	0b0	0b0: treat UDP as 8 bits 0b1: treat UDP as 24 bits
0x337	ctrl1_mode_UDT	5	0b0	0b0: treat UDP as 8 bits 0b1: treat UDP as 24 bits
0x31C–0x31F	soft_bpp{x,y,z,u}	4:0	0b00000	Software override value
0x31C–0x31F	soft_bpp{x,y,z,u}_en	5	0b0	0b0: Software override disabled 0b1: Software override enabled

ADI Confidential for New Vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

43.3.4.5.2 Zero Padding

Pixel data received on a CSI-2 serializer can be zero-padded as it enters a video pipe up to a resulting bpp of 16. With zero padding, an input with multiple bpp rates can be routed to a single video pipe if the following conditions are met:

1. 8 ≤ bpp ≤ 16 for all incoming bpp rates that are routed to the video pipe.
    a. Zero padding occurs after doubling. The 8 ≤ bpp ≤ 16 requirement applies to the resulting bpp after doubling.
2. Bandwidth is lost proportionally to the amount of zero padding. Some amount of GMSL2 bandwidth will be dedicated to sending zeros instead of the original CSI-2 data. Ensure system bandwidth requirements can be met by using the calculations shown in the Interface-Specific Bandwidth Calculations section.
3. The video pipe's PCLK drift detection must be disabled.

Zero padding is set on a per pipe basis and applies to all data routed to the pipe. When enabled, the pipe PCLK is set to the fastest incoming PCLK (i.e., smallest bpp), and all data within the pipe is treated as having a pixel width set by the BPP bitfield for that pipe. To enable zero padding, set AUTO_BPP = b0, BPP = largest bpp in the pipe (≤16), soft_bpp = smallest bpp in the pipe, and soft_bpp_en = b1. PCLK drift detection must also be disabled via the pipe's DRIFT_DET_EN bit. Using this method, all incoming datatypes with a bpp < BPP are zero padded so that all bpp rates within the pipe are equal. Reference Table 210 for zero padding registers.

The zero-padded data will automatically be recovered correctly on the deserializer based on the DT information that is automatically transmitted to the deserializer.

Table 210. GMSL2 CSI-2 Serializer Zero Padding Registers

Register Address (Video Pipe)	Bitfield Name	Bits	POR	Decode
0x100 (Pipe X) **0x108 (Pipe Y)** **0x110 (Pipe Z)** **0x118 (Pipe U)**	AUTO_BPP	3	0b1	0b0: Use bpp from bpp register 0b1: Use bpp from MIPI receiver
0x101 (Pipe X) **0x109 (Pipe Y)** **0x111 (Pipe Z)** **0x119 (Pipe U)**	BPP	5:0	0x18	Number of bits per pixel
0x102 (Pipe X) **0x10A (Pipe Y)** **0x11A (Pipe Z)** **0x119 (Pipe U)**	DRIFT_DET_EN	1	0b1	Enable PCLK frequency drift detection, resets video pipeline upon error and reports it
0x31C–0x31F	soft_bpp{x,y,z,u}	4:0	0b00000	Software override value
0x31C–0x31F	soft_bpp{x,y,z,u}_en	5	0b0	0b0: Software override disabled 0b1: Software override enabled

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
Analog Devices Internal Distribution Only

43.3.4.5.3 Double Mode and Zero Padding Example

EMB8 (DBL), RAW12, and RAW16 share pipe X. EMB8 is doubled to 16 bpp. RAW12 is zero padded to 16 bpp. RAW16 is unmodified. All datatypes will be 16 bpp inside of the pipe. EMB8 is doubled rather than zero padded because doubling is more efficient than zero padding, and EMB8 (DBL) has a bpp equal to the largest bpp in the pipe (RAW16).

- AUTO_BPP = b0 – Do not set bpp based on CSI-2 header (pipe X).
- BPP = 0x10 – Force pipe X bpp to 16 by zero-padding.
- soft_bppx = 0x0C – Must be set to the smallest input bpp (before padding and after doubling).
- soft_bppx_en = b1 – All data with DT 0x00 (FS) and VC0 that is in this pipe.
- bpp8dblx = b1 – All data with DT 0x00 (FS) and VC0 that is in this pipe.
- DRIFT_DET_EN = b0 – PCLK frequency drift detection is disabled for pipe X.

43.3.4.6 Disabling Heartbeat Mode (Serializer)

By default, all GMSL2 serializers send “heartbeat” packets during blanking intervals (see the Video Basics section and Interface-Specific Bandwidth Calculations). Heartbeat packets are GMSL packets that only contain low-frequency HVD signals (i.e., HS, VS, DE). Sending these packets during blanking ensures that the video clock regeneration in the receiver can properly track the number of pixel clocks that should be generated at the output. These packets are not required for CSI-2 data transmission (precise blanking timing is not required). By disabling the heartbeat, this bandwidth can be made available to other types of data.

Note: Heartbeat mode must be enabled in the GMSL2 CSI-2 serializer when connected to a non-CSI-2 deserializer.

Heartbeat mode is disabled independently for each video pipe by setting LIM_HEART = b1 in the serializer (Table 211). Reference Disabling Heartbeat Mode (Deserializer) for GMSL2 CSI-2 deserializer configuration details.

Note: The LIM_HEART bitfield has different functionality in the GMSL2 CSI-2 quad deserializer devices than other GMSL2 devices. In the CSI-2 quad deserializers, this bitfield defines the timeout to detect loss of video lock. See the GMSL2 CSI-2 Quad deserializer Heartbeat Mode section for more information.

Table 211. GMSL2 CSI-2 Serializer Heartbeat Disable Registers

Register Address (Video Pipe)	Bitfield Name	Bits	POR	Decode
0x102 (Pipe X) 0x10A (Pipe Y) 0x112 (Pipe Z) 0x11A (Pipe U)	LIM_HEART	2	0b0	Disable heartbeat during blanking. Use with SEQ_MISS_EN and DIS_PKT_DET registers in Deserializer. 0b0: Heartbeat enabled during blanking 0b1: Heartbeat disabled during blanking

ADI Confidential for New Visitor under NDA
Maxim Integrated Company Confidential Provided to GMSL2 Customers Under NDA

43.3.4.7 Pipe Tx Enable and Stream IDs

Each video pipe has a dedicated transmit enable bit that must be set for video data to be transmitted through the video pipe. GMSL2 CSI-2 serializers have pipe X and pipe Z enabled by default at power-up. Enable pipes that are required for the application; disable any pipes that are not in use. Reference Table 212 for relevant registers.

After data is transmitted through the video pipe(s), a stream ID is assigned prior to transmission across the serial link. The stream ID is then used by the deserializer to determine video pipe routing. Each video pipe has a dedicated stream ID that is set with the TX_STR_SEL[1:0] bitfield.

The MAX9295E has two GMSL links. The TX_SPLT_MASK_{A,B} bits are used to select which links(s) connect to a given video pipe. Video pipes can be connected to one or both GMSL links. By default, all video pipes send data to link A. Set TX_SPLT_MASK_B = 1 to enable sending data from a video pipe to link B.

Note: Stream IDs must be set in the serializer to avoid data Rx conflicts when using reverse splitter mode (i.e., multiple serializers connected to a single deserializer). See the GMSL2 Link Basics – Splitter Mode section for additional information.

Table 212. GMSL2 CSI-2 Serializer Pipe Enable and Stream ID Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x02	VID_TX_EN_{U,Z,Y,X}	7:4	0b0101	0b0: Video transmit disabled for the associated pipe 0b1: Video transmit enabled for the associated pipe
0x53	TX_STR_SEL[1:0]	1:0	0b00	Stream ID for packets transmitted from pipe X
0x53	TX_SPLT_MASK_A TX_SPLT_MASK_B	4 5	0b0 0b1	Pipe X routing to GMSL link. 0b0: Video transmit disabled for the associated link 0b1: Video transmit enabled for the associated link Note: Only applicable to MAX9295E
0x57	TX_STR_SEL[1:0]	1:0	0b01	Stream ID for packets transmitted from pipe Y
0x57	TX_SPLT_MASK_A TX_SPLT_MASK_B	4 5	0b0 0b1	Pipe Y routing to GMSL link. 0b0: Video transmit disabled for the associated link 0b1: Video transmit enabled for the associated link Note: Only applicable to MAX9295E
0x5B	TX_STR_SEL[1:0]	1:0	0b10	Stream ID for packets transmitted from pipe Z
0x5B	TX_SPLT_MASK_A TX_SPLT_MASK_B	4 5	0b0 0b1	Pipe Z routing to GMSL link. 0b0: Video transmit disabled for the associated link 0b1: Video transmit enabled for the associated link Note: Only applicable to MAX9295E
0x5F	TX_STR_SEL[1:0]	1:0	0b11	Stream ID for packets transmitted from pipe U
0x5F	TX_SPLT_MASK_A TX_SPLT_MASK_B	4 5	0b0 0b1	Pipe U routing to GMSL link. 0b0: Video transmit disabled for the associated link 0b1: Video transmit enabled for the associated link Note: Only applicable to MAX9295E

ADI Confidential for Maxim Integrated Company Confidential, Provided to GMSL2 Customers under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers under NDA

43.3.5 Status and Error Bits

43.3.5.1 MIPI CSI-2 PHY and Controller Errors

At the PHY layer, the MIPI Rx PHY module reports invalid low-power sequence errors for each clock and data lane. It also issues interrupts for high-speed SYNC pattern errors and the deskew calibration status.

At the protocol level, the CSI-2 Rx controller reports header ECC bit-errors, payload CRC errors, line length errors (based on the word count indicated in the packet header), and frame count errors.

MIPI PHY and controller error bits are logically ORed together to form the MIPI_ERR_FLAG, which issues an ERROR/ERRB interrupt by default.

See Table 213 for a list of CSI-2 serializer error registers.

Table 213. GMSL2 CSI-2 Serializer Error Registers

Register Address	Bitfield Name	Function	Decode
0x1F	MIPI_ERR_FLAG	MIPI Rx Error Flag	Asserted when any of the following is asserted: • phy0_hs_err [0],[1],[4],[5] • phy1_hs_err [0],[1],[4],[5] • phy2_hs_err [0],[1],[4],[5] • phy3_hs_err [0],[1],[4],[5] • ctrl0_csi_err_l [0],[1],[7] • ctrl0_csi_err_h [0] • ctrl1_csi_err_l [0],[1],[7] • ctrl1_csi_err_h [0]
0x1E	MIPI_ERR_OEN		Enables reporting of MIPI Rx errors (MIPI_ERR_FLAG - 0x1F) at ERRB pin
0x339	phy0_lp_err	LP sequence errors of PHY0	0bxxxx1: Unrecognized Escape command received from data lane D0 0bxxx1x: Unrecognized Escape command received from CLK lane 0bxx1xx: Invalid line sequence detected from data lane D0 0bx1xxx: Invalid line sequence detected from data lane D1 0b1xxxx: Invalid line sequence detected from CLK lane
0x33B	phy1_lp_err	LP sequence errors of PHY1	0bxxxx1: Unrecognized Escape command received from data lane D0 0bxxx1x: Unrecognized Escape command received from CLK lane 0bxx1xx: Invalid line sequence detected from data lane D0 0bx1xxx: Invalid line sequence detected from data lane D1 0b1xxxx: Invalid line sequence detected from CLK lane
0x33D	phy2_lp_err	LP sequence errors of PHY2	0bxxxx1: Unrecognized Escape command received from data lane D0 0bxxx1x: Unrecognized Escape command received from CLK lane 0bxx1xx: Invalid line sequence detected from data lane D0 0bx1xxx: Invalid line sequence detected from data lane D1 0b1xxxx: Invalid line sequence detected from CLK lane
0x33F	phy3_lp_err	LP sequence errors of PHY3	0bxxxx1: Unrecognized Escape command received from data lane D0 0bxxx1x: Unrecognized Escape command received from CLK lane 0bxx1xx: Invalid line sequence detected from data lane D0 0bx1xxx: Invalid line sequence detected from data lane D1 0b1xxxx: Invalid line sequence detected from CLK lane
0x33A	phy0_hs_err	HS status and error of PHY0 data lanes	0bxxxxxxx1: HS sync pattern with 2 or more bit errors detected on data lane D0 0bxxxxxx1x: HS sync pattern with 2 or more bit errors detected on data lane D1 0bxxxx1xx: HS sync pattern with 1-bit error detected on data lane D0 0bxxx1xxx: HS sync pattern with 1-bit error detected on data lane D1 0bxxx1xxxx: High speed receiver skew calibration failed on data lane D1

ADI Confidential for Newsight under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_ Customers under NDA

			0bxx1xxxxx: High speed receiver skew calibration failed on data lane D0 0bx1xxxxxx: High speed receiver skew calibration run on data lane D1 0b1xxxxxxx: High speed receiver skew calibration run on data lane D0
0x33C	phy1_hs_err	HS status and error of PHY1 data lanes	0bxxxxxxx1: HS sync pattern with 2 or more bit errors detected on data lane D0 0bxxxxxx1x: HS sync pattern with 2 or more bit errors detected on data lane D1 0bxxxxx1xx: HS sync pattern with 1-bit error detected on data lane D0 0bxxxx1xxx: HS sync pattern with 1-bit error detected on data lane D1 0bxxx1xxxx: High speed receiver skew calibration failed on data lane D1 0bxx1xxxxx: High speed receiver skew calibration failed on data lane D0 0bx1xxxxxx: High speed receiver skew calibration run on data lane D1 0b1xxxxxxx: High speed receiver skew calibration run on data lane D0
0x33E	phy2_hs_err	HS status and error of PHY2 data lanes	0bxxxxxxx1: HS sync pattern with 2 or more bit errors detected on data lane D0 0bxxxxxx1x: HS sync pattern with 2 or more bit errors detected on data lane D1 0bxxxxx1xx: HS sync pattern with 1-bit error detected on data lane D0 0bxxxx1xxx: HS sync pattern with 1-bit error detected on data lane D1 0bxxx1xxxx: High speed receiver skew calibration failed on data lane D1 0bxx1xxxxx: High speed receiver skew calibration failed on data lane D0 0bx1xxxxxx: High speed receiver skew calibration run on data lane D1 0b1xxxxxxx: High speed receiver skew calibration run on data lane D0
0x340	phy3_hs_err	HS status and error of PHY3 data lanes	0bxxxxxxx1: HS sync pattern with 2 or more bit errors detected on data lane D0 0bxxxxxx1x: HS sync pattern with 2 or more bit errors detected on data lane D1 0bxxxxx1xx: HS sync pattern with 1-bit error detected on data lane D0 0bxxxx1xxx: HS sync pattern with 1-bit error detected on data lane D1 0bxxx1xxxx: High speed receiver skew calibration failed on data lane D1 0bxx1xxxxx: High speed receiver skew calibration failed on data lane D0 0bx1xxxxxx: High speed receiver skew calibration run on data lane D1 0b1xxxxxxx: High speed receiver skew calibration run on data lane D0
0x341	ctrl0_csi_err_l	Low byte of the errors reported by CSI-2 Controller 0	0bxxxxxxx1: 1-bit ECC error detected and corrected 0bxxxxxx1x: 2-bit ECC error detected and packet dropped 0bxYYYYYxx: YYYYY = bit position of the 1-bit error 0b1xxxxxx: CRC error detected and packet dropped
0x342	ctrl0_csi_err_h	High byte of the errors reported by CSI-2 Controller 0	0bxx1: Packets terminated early 0bx1x: Frame count error detected 0b1xx: Unsupported data type received
0x343	ctrl1_csi_err_l	Low byte of the errors reported by CSI-2 Controller 1	0bxxxxxxx1: 1-bit ECC error detected and corrected 0bxxxxxx1x: 2-bit ECC error detected and packet dropped 0bxYYYYYxx: YYYYY = bit position of the 1-bit error 0b1xxxxxx: CRC error detected and packet dropped
0x344	ctrl1_csi_err_h	High byte of the errors reported by CSI-2 Controller 1	0bx1: Packets terminated early 0b1x: Frame count error detected 0b1xx: Unsupported data type received

ADI Confidential for New Vision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

43.3.5.2 Video Pipe and Controller Status and Errors

Important video pipe status and error registers are presented in Table 214. These are relevant to the Advanced Use Case Example. Note that these are generic video pipe errors and common to all GMSL2 devices.

Table 214. GMSL2 CSI-2 Serializer Video Pipe and Controller Status and Error Bits

Register Address	Bitfield Name	Bits	POR	Decode
0x102	PCLKDET	7	0b0	Pipe X PCLK Detection
0x10A				Pipe Y PCLK Detection
0x112				Pipe Z PCLK Detection
0x11A				Pipe U PCLK Detection
0x102	DRIFT_ERR	6	0b0	Pipe X PCLK Drift Error Detection After the video pipeline starts, PCLK cannot drift more than a certain amount without restarting the subsystem.
0x10A				Pipe Y PCLK Drift Error Detection After the video pipeline starts, PCLK cannot drift more than a certain amount without restarting the subsystem.
0x112				Pipe Z PCLK Drift Error Detection After the video pipeline starts, PCLK cannot drift more than a certain amount without restarting the subsystem.
0x11A				Pipe U PCLK Drift Error Detection After the video pipeline starts, PCLK cannot drift more than a certain amount without restarting the subsystem.
0x102	OVERFLOW	5	0b0	Pipe X FIFO Overflow Flag Overflow occurs when video input throughput is too high, or BW allocation on GMSL2 link for video is insufficient
0x10A				Pipe Y FIFO Overflow Flag Overflow occurs when video input throughput is too high, or BW allocation on GMSL2 link for video is insufficient
0x112				Pipe Z FIFO Overflow Flag Overflow occurs when video input throughput is too high, or BW allocation on GMSL2 link for video is insufficient
0x11A				Pipe U FIFO Overflow Flag Overflow occurs when video input throughput is too high, or BW allocation on GMSL2 link for video is insufficient
0x102	FIFO_WARN	4	0b0	Pipe X FIFO is more than half full
0x10A				Pipe Y FIFO is more than half full
0x112				Pipe Z FIFO is more than half full
0x11A				Pipe U FIFO is more than half full
0x3D2-0x3D7	ctrl{0,1}_csi_dt_rcv{0-2}	7:0	0x00	The last 3 received datatypes are reported for each controller

ADI Confidential for Review Under NDA
Analog Devices, Inc.
Maxim Integrated Company Confidential
Provided to GMSL2_ Customer_ under NDA

43.4 Deserializer Configuration

The GMSL2 CSI-2 deserializer must be configured before receiving input from the serial link. Dynamic configuration is not supported.

1. Program video pipes and routing from GMSL core.
2. Configure the crossbar switch and bpp/DT/VC override.
3. Configure pipe-to-controller routing.
4. Configure MIPI port lane mapping and polarity inversion (if necessary).
5. Configure the MIPI D-PHY Tx.
6. Adjust deskew and PHY timing settings.
7. Configure data replication (if necessary).

Note: Data will not be output from the deserializer while the serializer and deserializer are locking to the timing of the incoming video. This can result in the absence of a frame start and missing lines of data from the first frame output from the deserializer.

43.4.1 GMSL Core and Video Pipes

GMSL2 CSI-2 deserializers feature video pipes that transmit received data from the GMSL core to the CSI-2 Tx controller(s). Video data transmitted across the serial link is received by the deserializer's GMSL core and routed to the video pipe(s) according to stream ID. Watermark detection is only supported by pipe X. Data routing from the GMSL core to the CSI-2 Tx controllers is illustrated in Figure 198; stream ID configuration registers are presented in Table 215. Multiple video pipes can be assigned the same stream ID, however, video pipes may only output to one CSI-2 Tx controller.

Note: Assigning the same stream ID to multiple video pipes is a method of data replication. See the *Data Replication* section for additional information.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Information Provided under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

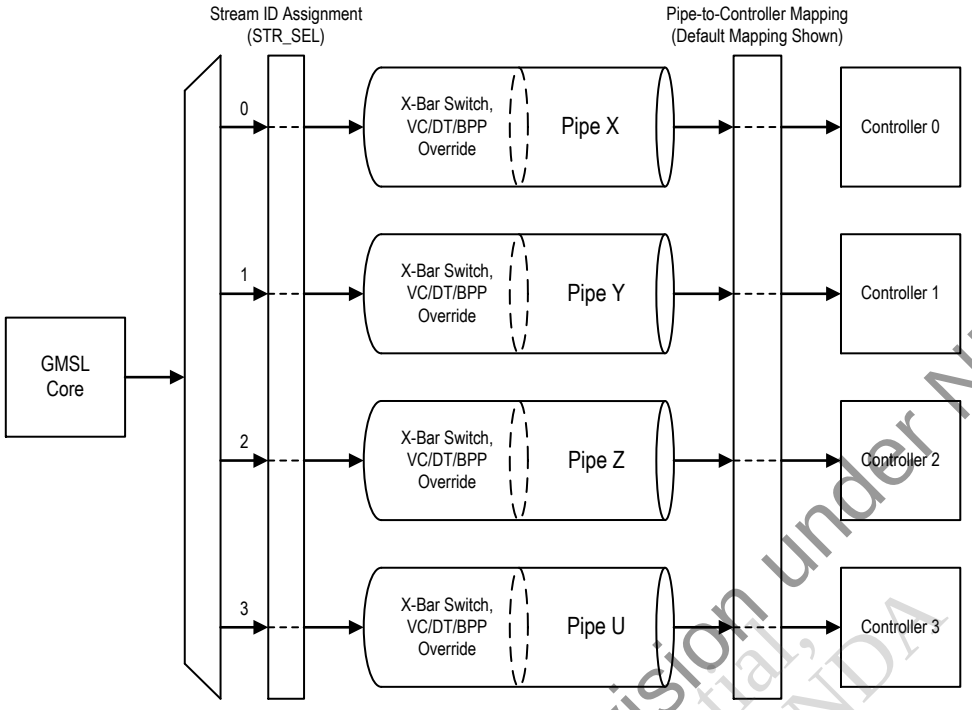


Figure 198. MAX9296 Pipe to Controller Routing

Table 215. GMSL2 CSI-2 Deserializer Video Pipe Stream ID Configuration Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x50	STR_SEL[1:0]	1:0	0b00	Pipe X stream select
0x51	STR_SEL[1:0]	1:0	0b01	Pipe Y stream select
0x52	STR_SEL[1:0]	1:0	0b10	Pipe Z stream select
0x53	STR_SEL[1:0]	1:0	0b11	Pipe U stream select

43.4.1.1 Video Crossbar

Video data passes through a video crossbar before entering the video pipe's line buffer. Each video pipe's crossbar can be used to rearrange the order of bits within a pixel, to force the state of a bit to 1 or 0 (including SYNC signals), or to invert the bit. Reference *Table 216* for VS crossbar registers.

In use cases with a CSI-2 source, the video crossbar is used primarily when data from two different video pipes are aggregated to a single deserializer output port, but the data exists within the same frame (e.g., embedded data and pixel data). Each video pipe includes a set of video SYNC signals. Since only one set of SYNC signals should be routed to the output when data is combined into a single frame, the SYNC signals in one of the video pipes can be forced to zero using the crossbar switch.

Table 216. GMSL2 CSI-2 Deserializer VS Crossbar Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x1D9 (CROSS_VS)	CROSS_VS_F	5	0b0	Force VSYNC to zero in pipe X
0x1F9 (CROSS_VS)	CROSS_VS_F	5	0b0	Force VSYNC to zero in pipe Y
0x219 (CROSS_VS)	CROSS_VS_F	5	0b0	Force VSYNC to zero in pipe Z
0x239 (CROSS_VS)	CROSS_VS_F	5	0b0	Force VSYNC to zero in pipe U

Analog Devices Confidential for New Revision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

43.4.1.2 BPP/DT/VC Override

Each video pipe can override the bits per pixel (bpp), datatype (DT) header, and virtual channel (VC) of the video data received from the serializer (Table 217). This is required when the data in the video pipe did not come from a CSI-2 source (i.e., there is no DT/VC associated with the source data). Rarely does the bpp sent from the serializer need to be overridden. The bpp, DT, and VC override changes the value as the data enters the pipe (before arriving at the pipe-to-controller routing block). These overrides are configured on a per pipe basis, and the override applies to all data within the configured pipe. There should only be a single video datatype within the pipe when using this feature. BPP, DT, and VC are all enabled with a single bit; all three fields must be configured correctly (soft_bpp, soft_dt, and soft_vc).

Alternatively, the data type and/or virtual channel can be selectively overridden using the MAP_SRC and MAP_DST registers in the pipe-to-controller mapping block (Table 218). Using the MAP_SRC and MAP_DST method is recommended when the source is CSI-2 but the DT/VC needs to be changed for aggregation (e.g., data from two serializers with identical CSI-2 sources are aggregated onto a single deserializer output port).

Table 217. GMSL2 CSI-2 Deserializer BPP, DT, and VC Override Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x31D	override_bpp_vc_dtx	6	0b0	Enable software override for BPP, VC, and DT in pipe X
0x31D	override_bpp_vc_dty	7	0b0	Enable software override for BPP, VC, and DT in pipe Y
0x320	override_bpp_vc_dtz	6	0b0	Enable software override for BPP, VC, and DT in pipe Z
0x320	override_bpp_vc_dtu	7	0b0	Enable software override for BPP, VC, and DT in pipe U
0x313	soft_bpp_x[4:0]	7:3	0b00000	Software defined BPP for pipe X
0x314	soft_vc_x[3:0]	3:0	0b0000	Software defined virtual channel for pipe X
0x314	soft_vc_y[3:0]	7:4	0b0000	Software defined virtual channel for pipe Y
0x315	soft_vc_z[3:0]	3:0	0b0000	Software defined virtual channel for pipe Z
0x315	soft_vc_u[3:0]	7:4	0b0000	Software defined virtual channel for pipe U
0x316	soft_dt_x[5:0]	5:0	0b00000	Software defined virtual channel for pipe X
0x316	soft_dt_y_h[1:0]	7:6	0b00	Software defined virtual channel for pipe Y (2 MSBs)
0x317	soft_dt_y_l[3:0]	3:0	0b0000	Software defined virtual channel for pipe Y (4 LSBs)
0x317	soft_dt_z_h[3:0]	7:4	0b0000	Software defined virtual channel for pipe Z (4 MSBs)
0x318	soft_dt_z_l[1:0]	1:0	0b00	Software defined virtual channel for pipe Z (2 LSBs)
0x318	soft_dt_u[5:0]	7:2	0b000000	Software defined virtual channel for pipe U
0x319	soft_bpp_y[4:0]	4:0	0b0000	Software defined BPP for pipe Y
0x319	soft_bpp_z_h[2:0]	7:5	0b000	Software defined BPP for pipe Z (3 MSBs)
0x31A	soft_bpp_z_l[1:0]	1:0	0b00	Software defined BPP for pipe Z (2 LSBs)
0x31A	soft_bpp_u[4:0]	6:2	0b00000	Software defined BPP for pipe U

ADI Confidential for Newsign under NDA
Analog Devices Inc. Maxim Integrated Company Confidential Provided to GMSL_Customers_Only under NDA

43.4.2 Pipe-to-Controller Mapping

Pipe-to-controller mapping is dependent on system design and requirements. This includes the number of input ports, datatypes (and bpp), and output ports in addition to data aggregation and duplication requirements.

Video data is mapped from the video pipes to the four CSI-2 Tx controllers. Each video pipe has dedicated register address space for pipe-to-controller mapping. This is used to aggregate data from multiple video pipes to a single CSI-2 output port. A video pipe can only be routed to a single controller. The configuration registers are contained in Table 218.

The pipe-to-controller mapping registers can be used to overwrite any previously assigned virtual channel to data within a video pipe. This is done by setting the 2 MSBs in the MAP_SRC register to the existing virtual channel then setting the 2 MSBs in the MAP_DST register to the target virtual channel. See the BPP/DT/VC Override section for alternative configurations.

Note: Data replication cannot be implemented using the pipe-to-controller mapping registers (see the Data Replication section). However, video streams can be combined using the mapping registers (see the Aggregation section).

Pipe-to-controller register space:

- Pipe X: MIPI_TX 0
- Pipe Y: MIPI_TX 1
- Pipe Z: MIPI_TX 2
- Pipe U: MIPI_TX 3

Default Mapping:

- Pipe X → Controller 0
- Pipe Y → Controller 1
- Pipe Z → Controller 2
- Pipe U → Controller 3

Table 218. GMSL2 CSI-2 Deserializer Pipe-to-Controller Mapping

Register Name	Bitfield Name	Bits	POR	Decode
MIPI_TX11	MAP_EN_L[7:0]	7:0	0x00	Mapping enable low byte [7:0]. Each bit enables 1 of 8 mapping and distribution entries (defined in MAP_SRC_x, MAP_DST_x and MAP_DPHY_DST_x) for the current video stream. Non-matched virtual channel (VC) and data types (DT) pass to the corresponding CSI2 controller. 0bxxxxxxx1: Enable MAP_SRC_0 and MAP_DST_0 registers 0bxxxxxx1x: Enable MAP_SRC_1 and MAP_DST_1 registers 0bxxxxx1xx: Enable MAP_SRC_2 and MAP_DST_2 registers 0bxxxx1xxx: Enable MAP_SRC_3 and MAP_DST_3 registers 0bxxx1xxxx: Enable MAP_SRC_4 and MAP_DST_4 registers 0bxx1xxxxx: Enable MAP_SRC_5 and MAP_DST_5 registers 0bx1xxxxxx: Enable MAP_SRC_6 and MAP_DST_6 registers 0b1xxxxxxx: Enable MAP_SRC_7 and MAP_DST_7 registers
MIPI_TX12	MAP_EN_H[7:0]	7:0	0x00	Mapping enable high byte [15:8]. Each bit enables 1 of 8 mapping and distribution entries (defined in MAP_SRC_x, MAP_DST_x and MAP_DPHY_DST_x) for the current video stream. Non-matched virtual channel (VC) and data types (DT) pass to the corresponding CSI2 controller.

ADI Confidential for Newvision under NDA
ADI Confidential, Distribution limited to Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

				0bxxxxxxx1: Enable MAP_SRC_8 and MAP_DST_8 registers 0bxxxxx1x: Enable MAP_SRC_9 and MAP_DST_9 registers 0bxxxx1xx: Enable MAP_SRC_10 and MAP_DST_10 registers 0bxxx1xxx: Enable MAP_SRC_11 and MAP_DST_11 registers 0bxx1xxxx: Enable MAP_SRC_12 and MAP_DST_12 registers 0bx1xxxxx: Enable MAP_SRC_13 and MAP_DST_13 registers 0b1xxxxxx: Enable MAP_SRC_14 and MAP_DST_14 registers 0b1xxxxxx: Enable MAP_SRC_15 and MAP_DST_15 registers
MIPI_TX13-43 (odd registers)	MAP_SRC_{0-15}	7:0	0x00	Source DT and VC (present in the corresponding pipe) that will be mapped to the corresponding destination DT and VC and routed to the PHY configured in MIPI_TX45-48. Bit [7:6]: Virtual Channel Bit [5:0]: Data type
MIPI_TX14-44 (even registers)	MAP_DST_{0-15}	7:0	0x00	Destination DT and VC that will be routed to the PHY configured in MIPI_TX45-48. Bit [7:6]: Virtual Channel Bit [5:0]: Data type
MIPI_TX45	MAP_DPHY_DEST_{0-3}[1:0]	7:6 5:4 3:2 1:0	0x00	CSI2 Controller Destination for the corresponding MAP_SRC_x and MAP_DST_x registers 0b00: Map to Controller 0 0b01: Map to Controller 1 0b10: Map to Controller 2 0b11: Map to Controller 3
MIPI_TX46	MAP_DPHY_DEST_{4-7}[1:0]	7:6 5:4 3:2 1:0	0x00	CSI2 Controller Destination for the corresponding MAP_SRC_x and MAP_DST_x registers 0b00: Map to Controller 0 0b01: Map to Controller 1 0b10: Map to Controller 2 0b11: Map to Controller 3
MIPI_TX47	MAP_DPHY_DEST_{8-11}[1:0]	7:6 5:4 3:2 1:0	0x00	CSI2 Controller Destination for the corresponding MAP_SRC_x and MAP_DST_x registers 0b00: Map to Controller 0 0b01: Map to Controller 1 0b10: Map to Controller 2 0b11: Map to Controller 3
MIPI_TX48	MAP_DPHY_DEST_{12-15}[1:0]	7:6 5:4 3:2 1:0	0x00	CSI2 Controller Destination for the corresponding MAP_SRC_x and MAP_DST_x registers 0b00: Map to Controller 0 0b01: Map to Controller 1 0b10: Map to Controller 2 0b11: Map to Controller 3

43.4.2.1 Aggregation

Video streams can be combined and/or split onto multiple output ports using the connected serializer's controller-to-pipe mapping in combination with the deserializer's pipe-to-controller mapping (Table 218).

GMSL2 CSI-2 devices support first come, first serve (FCFS) aggregation. Multiple video pipes can be routed to the same output port using the deserializer mapping registers. This does not require that data come from the same source. Note that the data is output from the controller based on which complete packet is in memory first.

ADI Confidential for New Wish under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential,
Provided to GMSL-Customer under NDA

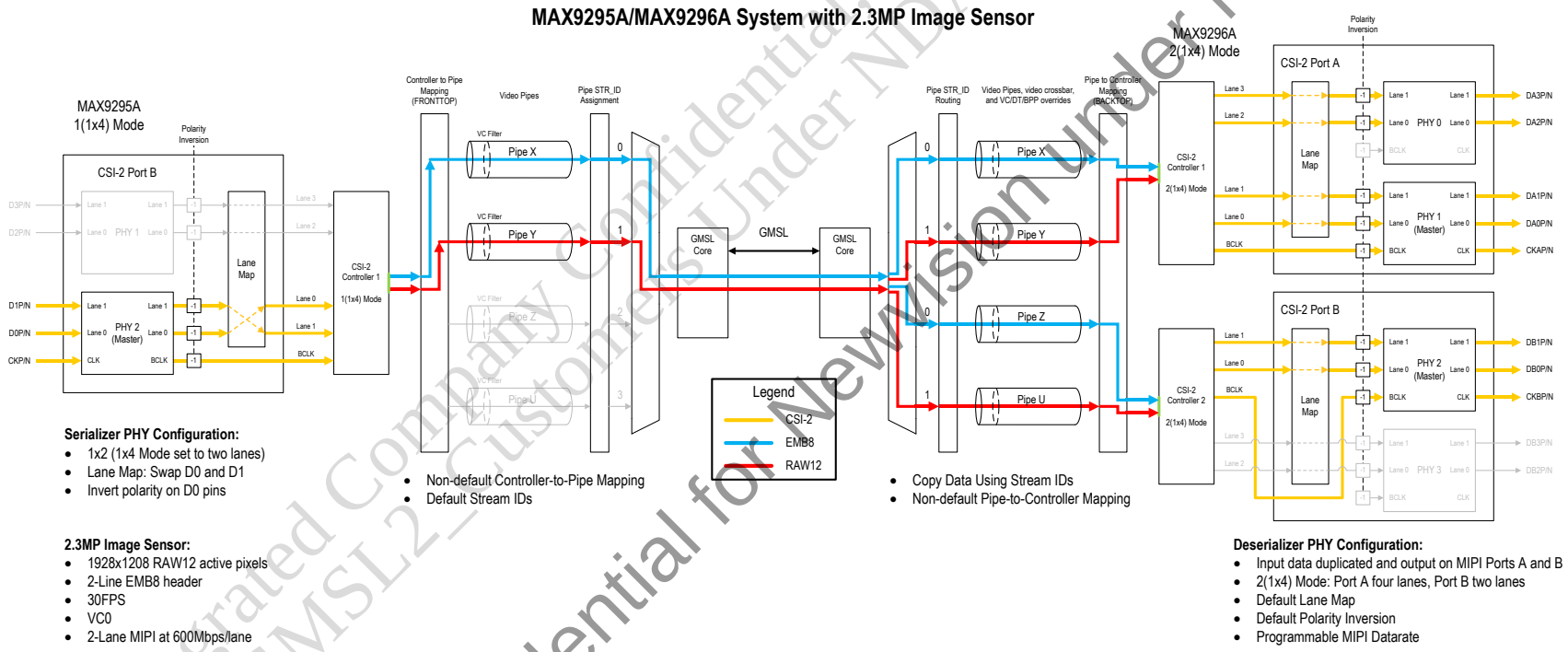


Figure 199. MAX9295A/MAX9296A Using 2.3MP Image Sensor (Advanced Use Case)

Note: Reference the GMSL2 CSI-2 Advanced Use Case Example below for more information regarding the use case presented in Figure 199.

Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
ADI Confidential for NewVision under NDA

43.4.2.1.1 Pipe-to-Controller Mapping – Advanced Use Case Example

The deserializer pipe-to-controller routing shown in Figure 199 (see the GMSL2 CSI-2 Advanced Use Case Example below for more information) is accomplished using the following register settings. Note that any data in a video pipe with DT and VC attributes that do not match the corresponding and enabled MAP_SRC registers will have the default pipe-to-controller routing (see above).

Pipe X (MIPI_TX 0 address space)

Route DT 0x12 (EMB8), DT 0x00 (Frame Start), and DT 0x01 (Frame End) to Controller 1 (Port A). All DTs in the pipe are VC0, this will not be changed by this mapping configuration.

- MAP_SRC_0 = 0x12 – All data with DT 0x12 and VC0 that is in this pipe
- MAP_DST_0 = 0x12 – will be output as DT 0x12 and VC0.
- MAP_SRC_1 = 0x00 – All data with DT 0x00 (FS) and VC0 that is in this pipe
- MAP_DST_1 = 0x00 – will be output as DT 0x00 (FS) and VC0.
- MAP_SRC_2 = 0x01 – All data with DT 0x01 (FE) and VC0 that is in this pipe
- MAP_DST_2 = 0x01 – will be output as DT 0x01 (FE) and VC0.
- MAP_EN_L = b0000 0111 – Enable MAP_SRC and MAP_DST registers 0-2.
- MAP_DPHY_DEST_0 = b01 – Route MAP_DST_0 to controller 1.
- MAP_DPHY_DEST_1 = b01 – Route MAP_DST_1 to controller 1.
- MAP_DPHY_DEST_2 = b01 – Route MAP_DST_2 to controller 1.

Pipe Y (MIPI_TX 1 address space)

Route DT 0x2C (RAW12) VC0 DT 0x00 to Controller 1 (Port A). This combines the data on pipe X and pipe Y and output the aggregate on Port A on a first come, first serve basis. Note that FS and FE exist on all pipes that contain pixel data. In this example use case, pipe X and pipe Y coexist within the same frame. To prevent duplicate FS and FE packets on the output, the VS is forced to zero on one of the pipes (see Short-Packet (VS) Processing for details).

- MAP_SRC_0 = 0x2C – All data with DT 0x2C and VC0 that is in this pipe
- MAP_DST_0 = 0x2C – will be output as DT 0x2C and VC0.
- MAP_EN_L = b0000 0001 – Enable MAP_SRC and MAP_DST registers 0.
- MAP_DPHY_DEST_0 = b01 – Route MAP_DST_0 to controller 1.
- CROSS_VS_F (VRX-Y address space) = b1 – Force pipe Y VS to zero.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Distribution Only Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

Pipe Z (MIPI_TX 2 address space)
The data in pipe Z is a duplicate of pipe X. The same configurations are used, except the destination is controller 2 (Port B).

- MAP_SRC_0 = 0x12 – All data with DT 0x12 and VC0 that is in this pipe
- MAP_DST_0 = 0x12 – will be output as DT 0x12 and VC0.
- MAP_SRC_1 = 0x00 – All data with DT 0x00 (FS) and VC0 that is in this pipe
- MAP_DST_1 = 0x00 – will be output as DT 0x00 (FS) and VC0.
- MAP_SRC_2 = 0x01 – All data with DT 0x01 (FE) and VC0 that is in this pipe
- MAP_DST_2 = 0x01 – will be output as DT 0x01 (FE) and VC0.
- MAP_EN_L = b0000 0111 – Enable MAP_SRC and MAP_DST registers 0-2.
- MAP_DPHY_DEST_0 = b10 – Route MAP_DST_0 to controller 2.
- MAP_DPHY_DEST_1 = b10 – Route MAP_DST_1 to controller 2.
- MAP_DPHY_DEST_2 = b10 – Route MAP_DST_2 to controller 2.

Pipe U (MIPI_TX 2 address space)
The data in pipe U is a duplicate of pipe Y. The same configurations are used, except the destination is controller 2 (Port B).

- MAP_SRC_0 = 0x2C – All data with DT 0x2C and VC0 that is in this pipe.
- MAP_DST_0 = 0x2C – Will be output as DT 0x2C and VC0.
- MAP_EN_L = b0000 0001 – Enable MAP_SRC and MAP_DST registers 0.
- MAP_DPHY_DEST_0 = b10 – Route MAP_DST_0 to controller 2.
- CROSS_VS_F (VRX U address space) = b1 – force pipe U VS to zero.

ADI Confidential for review under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers under NDA

43.4.2.2 Short-Packet (VS) Processing

Each video pipe contains its own set of SYNC signals (see the *CSI-2 Packets* section). A typical CSI-2 system requires only VS and DE signals. In the GMSL2 CSI-2 deserializer, the VS signals are used by the controllers to generate frame start (FS) and frame end (FE) packets. See the *VS Routing* section in the Serializer Configuration section for more information (the same *VS Routing Use Cases* are referenced below).

Each VS transition within a video pipe is associated with a VC (determined by either the VC/DT information sent from the serializer or the BPP/DT/VC Override in the deserializer). The VS signal must be routed to an output controller in the same way as pixel data. Frame start packets are DT 0x00; frame end packets are DT 0x01 (reference the Pipe-to-Controller Mapping – Advanced Use Case Example). If FS/FE signals exist in a video pipe and are not routed to an output, the buffer will overflow, resulting in corrupted output.

Case 1.b) Single Virtual Channel, Multiple Datatypes (Embedded Data)

If a virtual channel is split into multiple pipes (due to differences in bpp) on the serializer, the VS signal will be duplicated. When routing to the deserializer output, only one VS should be used to prevent duplicate FS/FE and buffer overflow conditions. This can be accomplished by:

- Forcing unused VS signals to zero (see the Video Crossbar section), or
- Routing unused VS signals to an unused MIPI controller using the MAP_DPHY_DEST bitfield (Table 218).

Case 2.a) Frame-Synchronized Line-Interleaved (Multiple VCs that share a VS)

The VS_VC{1-4}_{L,H} bits can be used to duplicate a single VS signal to create multiple FS/FE packets with unique VCs (*Table 219*). VS_VC{1-4}_{L,H} is a 16-bit word (low and high byte). The register name index indicates the pipe/controller the bitfield acts upon (i.e., VS_VC1_{L,H} is for data in Pipe X/Controller 1) and the bit position indicates the VC number of the FS/FE that will be created. Any combination of VCs can be simultaneously set. When VS_VC{1-4}_{L,H} is not 0x00, the original VC of the VS signal is overridden based on the VS_VC setting (i.e., if the original VC is 0, and VS_VC is set to output on VC1, FS/FE will only be generated for VC1). After setting the output VCs with VS_VC{1-4}_{L,H}, the FS/FE must be routed to an output controller using the pipe-to-controller registers. If FS/FE are generated using VS_VC1_{L,H} and are not routed to an output, the buffer will overflow.

CSI-2 deserializers have a register option to output the SYNC signals for each pipe to an MFP. This can be used while debugging pipe-to-controller mapping. See Table 226 for details.

Table 219. GMSL2 CSI-2 Deserializer Frame-Start/End Override

Register Address	Bitfield Name	Bits	POR	Decode
0x309	VS_VC1_L	7:0	0x00	Pipe X frame start and frame end count and VC override. Low byte (VC0-7).
0x30A	VS_VC1_H	7:0	0x00	Pipe X frame start and frame end count and VC override. High byte (VC8-15).
0x30B	VS_VC2_L	7:0	0x00	Pipe Y frame start and frame end count and VC override. Low byte (VC0-7).
0x30C	VS_VC2_H	7:0	0x00	Pipe Y frame start and frame end count and VC override. High byte (VC8-15).
0x30D	VS_VC3_L	7:0	0x00	Pipe Z frame start and frame end count and VC override. Low byte (VC0-7).
0x30E	VS_VC3_H	7:0	0x00	Pipe Z frame start and frame end count and VC override. High byte (VC8-15).
0x30F	VS_VC4_L	7:0	0x00	Pipe U frame start and frame end count and VC override. Low byte (VC0-7).
0x310	VS_VC4_H	7:0	0x00	Pipe U frame start and frame end count and VC override. High byte (VC8-15).

Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA
API Confidential for Review Only Under NDA

43.4.2.3 Double Mode (Deserializer)

Double mode in GMSL2 CSI-2 deserializers must be configured according to the programming of the connected serializer(s) to properly decode the doubled GMSL2 video and output the correct CSI-2 format. Mirroring the serializer, each deserializer video pipe has a double bit for bpp rates = 8, 10, and 12. Additionally, the controller receiving the doubled data (determined by Pipe-to-Controller Mapping) must also be configured with the alternative memory map settings (for doubled data). If ctrl0_mode_UDT and/or ctrl1_mode_UDT is set on the serializer, then the deserializer must have the MODE_DT bit set. See Table 220 for a list of deserializer double mode registers.

Note: The deserializer double mode settings must be configured to match the serializer double mode configuration. Reference the Double Mode (Serializer) section for additional information. The double mode enable bits are set for the pipe transmitting the doubled data, while the alternative memory map and MODE_DT are set in the destination controller.

Table 220. GMSL2 CSI-2 Deserializer Double Mode Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x31C	bpp8dbl{x,y,z,u}	7:4	0b0	BPP=8 processed as 16-bit color enable for pipeline
0x31F	bpp8dbl{x,y,z,u}_mode	7:4	0b0	Enable 8-bit write alternate map to RAMs for pipeline
0x327	bpp10dbl{x,y,z,u}	3:0	0b0	BPP=10 processed as 20-bit color enable for pipeline
0x327	bpp10dbl{x,y,z,u}_mode	7:4	0b0	Enable 10-bit write alternate map to RAMs for pipeline
0x328	bpp12dbl{x,y,z,u}	3:0	0b0	BPP=12 processed as 24-bit color enable for pipeline
0x433 (CTRL 0) **0x473 (CTRL 1)** **0x4B3 (CTRL 2)** **0x4F3 (CTRL 3)**	ALT_MEM_MAP12	0	0b0	Alternative memory map enable for 12-bit DT
0x433 (CTRL 0) **0x473 (CTRL 1)** **0x4B3 (CTRL 2)** **0x4F3 (CTRL 3)**	ALT_MEM_MAP8	1	0b0	Alternative memory map enable for 8-bit DT
0x433 (CTRL 0) **0x473 (CTRL 1)** **0x4B3 (CTRL 2)** **0x4F3 (CTRL 3)**	ALT_MEM_MAP10	2	0b0	Alternative memory map enable for 10-bit DT
0x433 (CTRL 0) **0x473 (CTRL 1)** **0x4B3 (CTRL 2)** **0x4F3 (CTRL 3)**	MODE_DT	3	0b0	MIPI TX enable 24-bit packing of 8-bit MIPI UDP
0x433 (CTRL 0) **0x473 (CTRL 1)** **0x4B3 (CTRL 2)** **0x4F3 (CTRL 3)**	ALT2_MEM_MAP8	4	0b0	Alternative memory map enable for 8-bit DT

43.4.2.3.1 Double Mode Example Configurations

The table below provides examples of the serializer and deserializer register writes required to configure Double Mode in a GMSL2 CSI-2 system (Table 221). Double mode settings are set for the video pipe that contains the data; alternative memory map settings are set in the destination controller.

Maxim ADI Confidential for NewVision under NDA
Provided to GMSL2_Customers Under NDA
Maxim Integrated Company Confidential
Maxim Confidential Property

Table 221. Example Double Mode Settings for a Serializer and Deserializer Pair

Data Type(s) in Pipe	Serializer Settings	Deserializer Settings	Notes
8 bit doubled	bpp8dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0x10 soft_bpp{?}_en = 0b1	bpp8dbl{X,Y,Z,U} = 0b1 bpp8dbl{X,Y,Z,U}_mode = 0b1 ALT_MEM_MAP8 = 0b1	
10 bit doubled	bpp10dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0x14 soft_bpp{X,Y,Z,U}_en = 0b1	ALT_MEM_MAP10 = 0b1	Native 20-bit data optional
12 bit doubled	bpp12dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0x18 soft_bpp{X,Y,Z,U}_en = 0b1	ALT_MEM_MAP12 = 0b1	Native 24-bit data optional
8 bit tripled	ctrl{0,1}_mode_UDT = 0b1 soft_bpp{X,Y,Z,U} = 0x18 soft_bpp{X,Y,Z,U}_en = 0b1	MODE_DT = 0b1	Supports EMB8 and DTs 0x31-0x37
8 bit doubled and 16 bit (RAW16, YUV8)	bpp8dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0x10 soft_bpp{?}_en = 0b1	ALT2_MEM_MAP8 = 0b1	
EMB8 tripled and RAW12 doubled	bpp12dbl{X,Y,Z,U} = 0b1 ctrl{0,1}_mode_UDT = 0b1 soft_bpp{X,Y,Z,U} = 0x18 soft_bpp{X,Y,Z,U}_en = 0b1	ALT_MEM_MAP12 = 0b1 MODE_DT = 0b1	
Zero-Padding Examples:			
For any input bpp ≤ 16	AUTO_BPP = 0b0 BPP = Maximum of all DTs DRIFT_DET_EN = b0 (VID_TX) soft_bpp{X,Y,Z,U} = Minimum of all DTs soft_bpp{X,Y,Z,U}_en = 0b1 **If there is an 8-bit input (DBL):** bpp8dbl{X,Y,Z,U} = 0b1	**If there is an 8-bit input (DBL):** ALT2_MEM_MAP8 = 0b1	Maximum BPP is after doubling and after zero-padding. Minimum BPP is after doubling but before zero-padding.
8 bit doubled with 10 bit zero padded to 16 bit	AUTO_BPP = 0b0 (VID_TX) BPP = 0x10 (VID_TX) DRIFT_DET_EN = b0 (VID_TX) bpp8dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0x0A soft_bpp{X,Y,Z,U}_en = 0b1	ALT2_MEM_MAP8 = 0b1	Native 16-bit data optional soft_bpp = minimum input bpp = 10 BPP = maximum bpp = 16
8 bit doubled with 12 bit zero padded to 16 bit	AUTO_BPP = 0b0 (VID_TX) BPP = 0x10 (VID_TX) DRIFT_DET_EN = b0 (VID_TX) bpp8dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0x0C soft_bpp{X,Y,Z,U}_en = 0b1	ALT2_MEM_MAP8 = 0b1	Native 16-bit data optional soft_bpp = minimum input bpp = 12 BPP = maximum bpp = 16
8 bit doubled with 10 and 12 bit zero padded to 16 bit	AUTO_BPP = 0b0 (VID_TX) BPP = 0x10 (VID_TX) DRIFT_DET_EN = b0 (VID_TX) bpp8dbl{X,Y,Z,U} = 0b1 soft_bpp{X,Y,Z,U} = 0xA soft_bpp{X,Y,Z,U}_en = 0b1	ALT2_MEM_MAP8 = 0b1	native 16-bit data optional soft_bpp = minimum input bpp = 10 BPP = maximum bpp = 16

ADI Confidential for New Revision under NDA
Maxim Integrated Products, Inc. Company Confidential, Provided as MSL_Customer Under NDA

43.4.2.4 Disabling Heartbeat Mode (Deserializer)

In the MAX9296 devices, heartbeat mode is disabled independently for each video pipe by setting SEQ_MISS_EN = b0 and DIS_PKT_DET = b1 (Table 222). The deserializer VID_LOCK bit indicates that at least 16 video packets have been received within a moving time window. When heartbeat is disabled, the VID_LOCK window is increased by ~100x. Reference Disabling Heartbeat Mode (Serializer) for additional information.

Note: Heartbeat mode must be enabled when using a non-CSI-2 deserializer.

Note: The LIM_HEART bitfield has different functionality in the CSI-2 quad deserializer devices than other GMSL2 devices. In the CSI-2 quad deserializers, this bitfield defines the timeout to detect loss of video lock. Reference the CSI-2 Quad Deserializers section for additional information.

Table 222. GMSL2 CSI-2 Deserializer Heartbeat Disable Registers (MAX9296 Only)

Register Address	Bitfield Name	Bits	POR	Decode
0x100 (Pipe X)	DIS_PKT_DET	4	0b0	Disable packet detector. If the video is restarted with a different BPP when the packet detector is disabled, toggle this register or the video receive enable register to make sure the video link restarts. 0b0: Packet detector enabled 0b1: Packet detector disabled
0x112 (Pipe Y)				
0x124 (Pipe Z)				
0x136 (Pipe U)				
0x100 (Pipe X)	SEQ_MISS_EN	4	0b1	Video sequence miss detection enable. 0b0: Disabled 0b1: Enabled
0x112 (Pipe Y)				
0x124 (Pipe Z)				
0x136 (Pipe U)				

Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA
ADI Confidential for Newvision under NDA

43.4.3 D-PHY Tx Configuration

The PHYs are grouped to form the MIPI port(s). Each port must have one (active) clock lane, though the number of data lanes is configurable. When multiple PHYs are combined to form a single port, the PHY that outputs the MIPI clock is referred to as the "master" PHY.

All GMSL2 CSI-2 deserializers have four MIPI D-PHYs (i.e., PHY0–3) that each control two data lanes; however, clock output capabilities differ by device package. The MAX9296A has clock outputs on PHY 1 and PHY2; the MAX9296B has clock outputs on all PHYs.

Description of MIPI port modes:

- **2x4 Mode**: Two independent four-lane MIPI ports are used. PHY0 and PHY1 are combined to form a four-lane port with the clock output from PHY1; PHY2 and PHY3 are combined to form a four-lane port with the clock output from PHY2 (Figure 200).
- **4x2 Mode**: PHY0–3 are used as four independent two-lane MIPI ports with each port outputting a clock (Figure 201).
- **1x4A + 2x2 Mode**: Three independent MIPI ports are used. PHY0 and PHY1 are combined to form a four-lane MIPI port with the clock output from PHY1. Two independent two-lane ports are formed from PHY2 and PHY3 with each outputting a clock (Figure 202).
- **1x4B + 2x2 Mode**: Three independent MIPI ports are used. Two independent two-lane ports are formed from PHY0 and PHY1 with each outputting a clock. PHY2 and PHY 3 are combined to form a four-lane MIPI port with the clock output from PHY2 (Figure 203).

Note: Unused deserializer MIPI PHYs should be disabled using the PHY_STDBY bitfields in the MIPI_PHY register block.

The CSI-2 deserializer variants have specific MIPI Port architecture. The port mode is set in register 0x330. Refer to Table 223 for the PHYs and MIPI port modes available to each part number. Configuration registers are presented in Table 224.

Table 223. CSI-2 Deserializers: MIPI Ports by Part Number

Part Number	PHYs Available	MIPI Clock Outputs	MIPI Port Modes
MAX9296A	PHY0–3	PHY1 and PHY2	2x4
MAX9296B	PHY0–3	PHY0–3	2x4, 4x2, 1x4(A/B) + 2x2

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

Table 224. GMSL2 CSI-2 Deserializer Port Configuration Registers

Register Address	Bitfield Name	Bits	POR	Decode
0x313	CSI_OUT_EN	1	0b1	Enable CSI output
0x330	phy_4x2	0	MAX9296A: 0b0 MAX9296B: 0b1	MIPI Output Configured as four two-lane MIPI ports
0x330	phy_2x4	2	MAX9296A: 0b1 MAX9296B: 0b0	MIPI Output Configured as two four-lane MIPI ports
0x330	phy_1x4a_22	3	0b0	MIPI output configured as three ports. Port A is a four data-lane port using PHY0 and PHY1. Port E and Port F are two data-lane ports using PHY2 and PHY3.
0x330	phy_1x4b_22	4	0b0	MIPI output configured as three ports. Port C and Port D are two two-lane ports, using PHY0 and PHY1. Port B is a four-lane port using PHY2 and PHY3.
0x332	phy_Stdby	4	0b1111	0b1111: All PHYs not in standby 0bXXX0: Put PHY0 in standby mode 0bXX0X: Put PHY1 in standby mode 0bX0XX: Put PHY2 in standby mode 0b0XXX: Put PHY3 in standby mode

43.4.3.1 CSI-2 Deserializer Port Architecture

Both the MAX9296A and MAX9296B have four MIPI D-PHYs, each of which control two data lanes. The MAX9296A can output clock signals from PHY1 and PHY2 and supports two independent four-lane ports (i.e., Port A and Port B). The MAX9296B can output clock signals from all four PHYs. This architecture supports four independent two-lane ports (i.e., Ports C, D, E, and F) as well as the combination of PHY0 and PHY1 and/or PHY2 and PHY3 to create four-lane ports (i.e., Port A and Port B). When the MAX9296B is configured to use a four-lane port, the clock is driven from PHY1 for Port A and PHY2 for Port B (similar to the MAX9296A). See Figure 200 through Figure 203 below for the available MIPI port configurations.

ADI Confidential Preview version under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

43.4.3.2 MIPI Output Data Rate

The MIPI output data rate is independently configurable for each deserializer output port. Reference the MAX9296A/B Deserializer D-PHY Rx diagrams below (Figure 200, Figure 201, Figure 202, Figure 203) to determine which PHY to configure based on the desired output port.

The bitrates are set in 100Mbps increments with the phy{0-4}_csi_tx_dpll_predef_freq bitfields, where 0x01 = 100Mbps/lane and 0x0F = 1500Mbps/lane. Note that the applicable phy{0-4}_csi_tx_dpll_fb_fraction_predef_en bitfield must be set to enable the 100Mbps step setting. When multiple PHYs are combined to create an output port, the clock speed is determined by the master PHY. For stability enhancement, it is recommended to always put the DPLL blocks in reset (config_soft_rst_n = 0) before changing the MIPI lane rate

Note: The output data rate can be set at with more granularity by using custom DPLL settings. See the *RCLKOUT Configuration* section and software configuration tool for more information.

The deserializer output data rate can be set independently of the serializer input data rate provided that the output data rate is set fast enough to maintain the minimum blanking requirements. Reference the Interface-Specific Bandwidth Calculations section for detailed calculations. If the output data rate is not sufficient, the FIFO overflow flag will assert in the deserializer, and the data will be dropped. It is recommended to set the output rate to the lowest setting that supports video throughput. The frames per second (fps) will be maintained as the data rate is increased beyond the incoming bandwidth requirements.

The minimum horizontal blanking period required by CSI-2 serializers and deserializers is either 40 pixels or 300ns + 370UI (where UI is defined as the period of the CSI-2 lane rate), whichever is greatest. Typically, 40 pixels is the larger figure. The minimum vertical blanking period and minimum vertical front porch are both one video line. The recommended vertical back porch is one video line.

Note: D-PHY uses a double data rate (DDR) clock. The clock frequency will be one half the bitrate/lane. The deserializer sets the output by Mbps/lane, not the clock frequency.

ADI Confidential for New Vision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

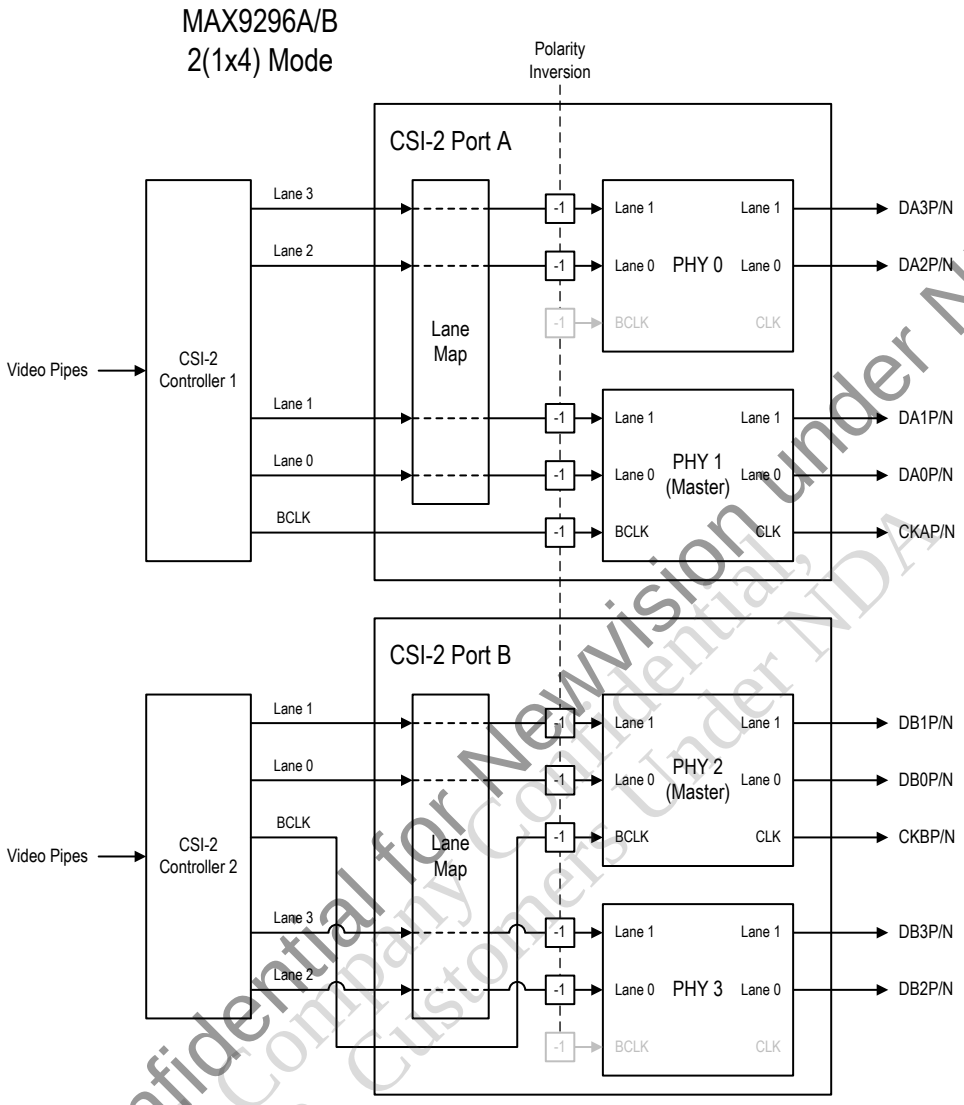


Figure 200. MAX9296A/B Deserializer D-PHY Rx – 2(1x4) Mode

Note: The MAX9296A and MAX9296B can both be configured for 2(1x4) Mode (Figure 200). The MAX9296A has clock outputs on PHY1 and PHY2. The greyed clock outputs on PHY0 and PHY3 represent the unused clock outputs that are available on the MAX9296B.

ADI Confidential for Newvision under NDA
Analog Devices Integrated Company Confidential, Provided to GMSL2_ Customers Under NDA
Maxim Integrated
Provided to GMSL2_

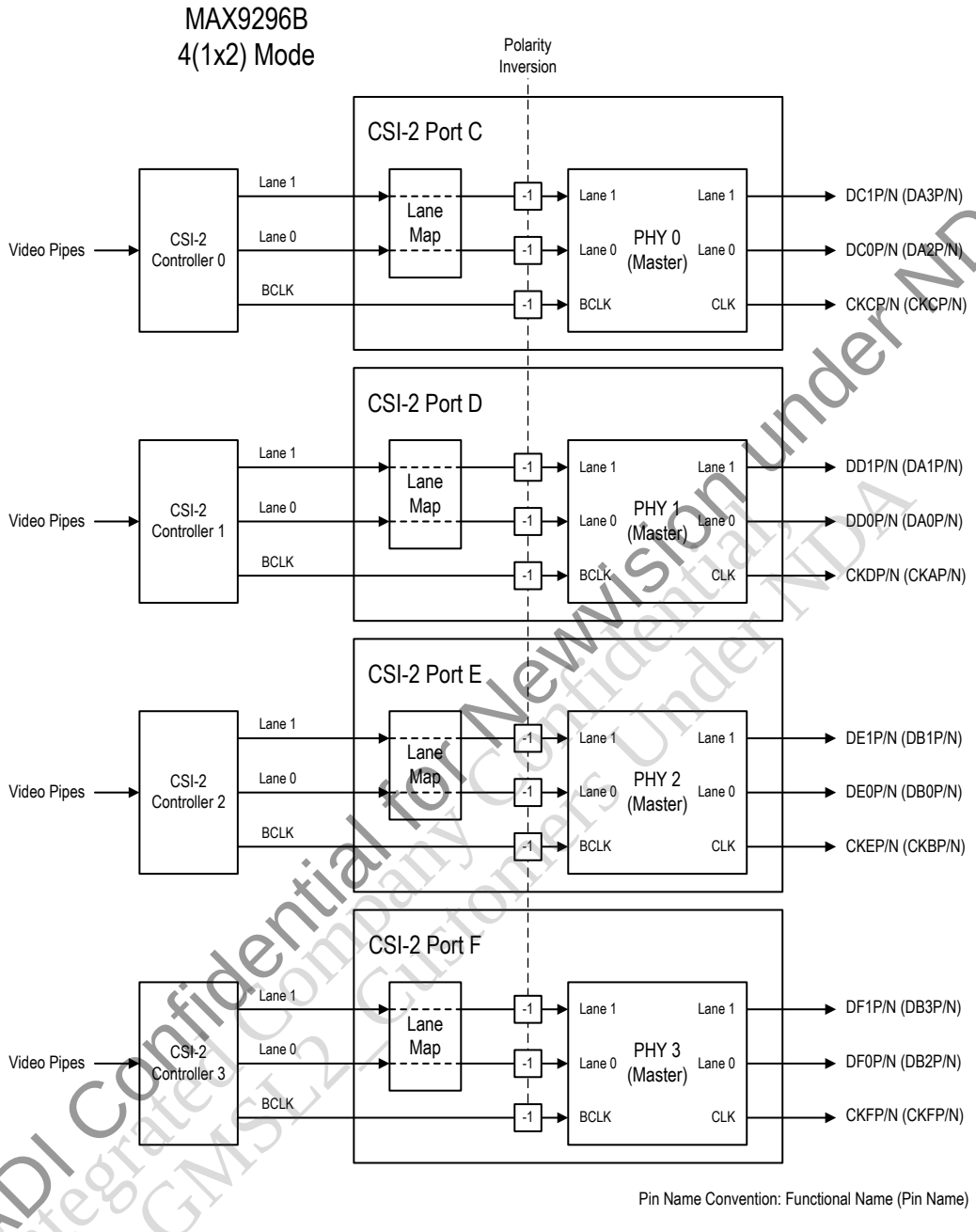


Figure 201. MAX9296A/B Deserializer D-PHY Rx – 4(1x2) Mode

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

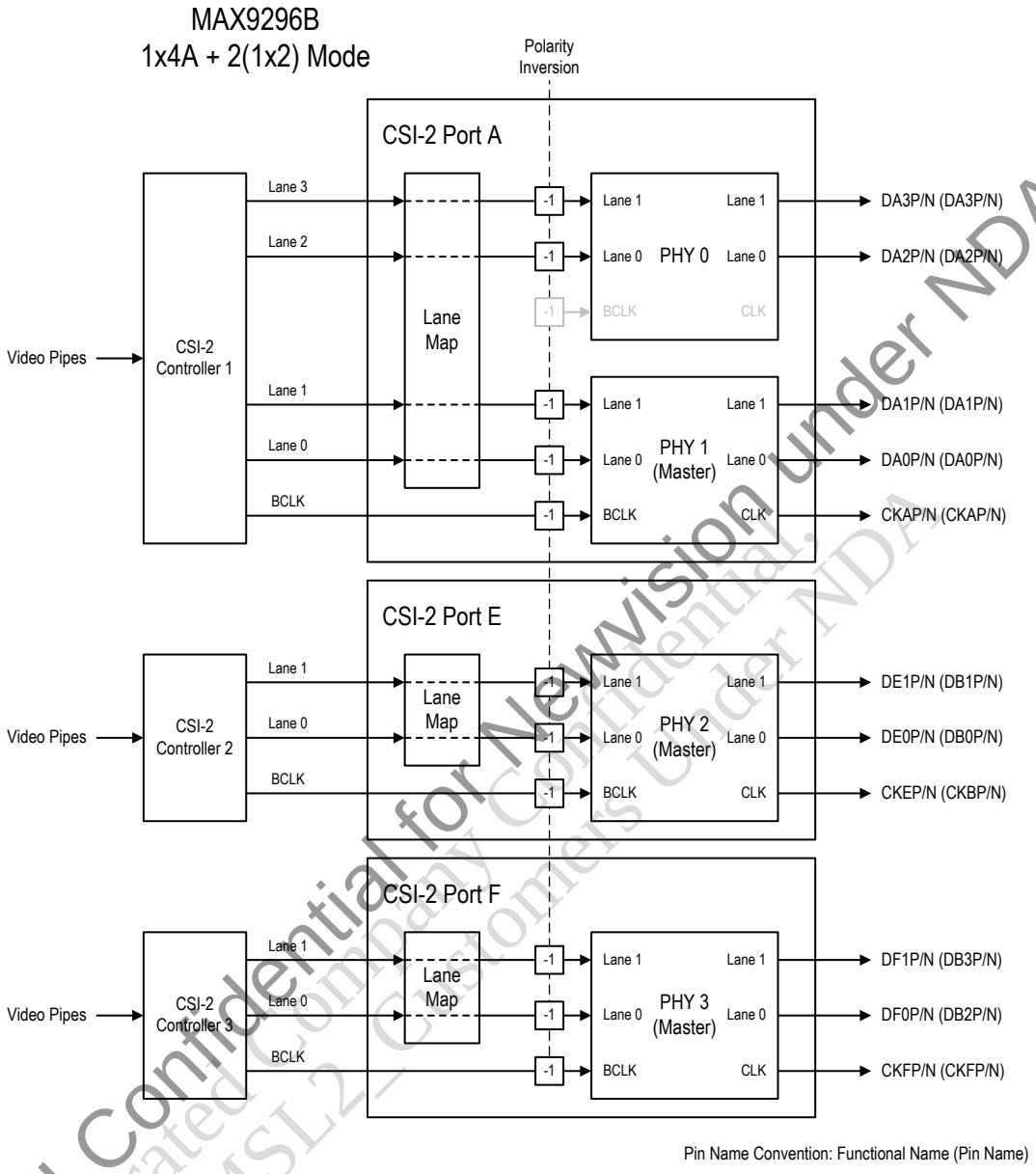


Figure 202. MAX9296B Deserializer D-PHY Rx – 1x4A + 2(1x2) Mode

ADI Confidential for Newvision under NDA
Analog Devices, Inc. Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

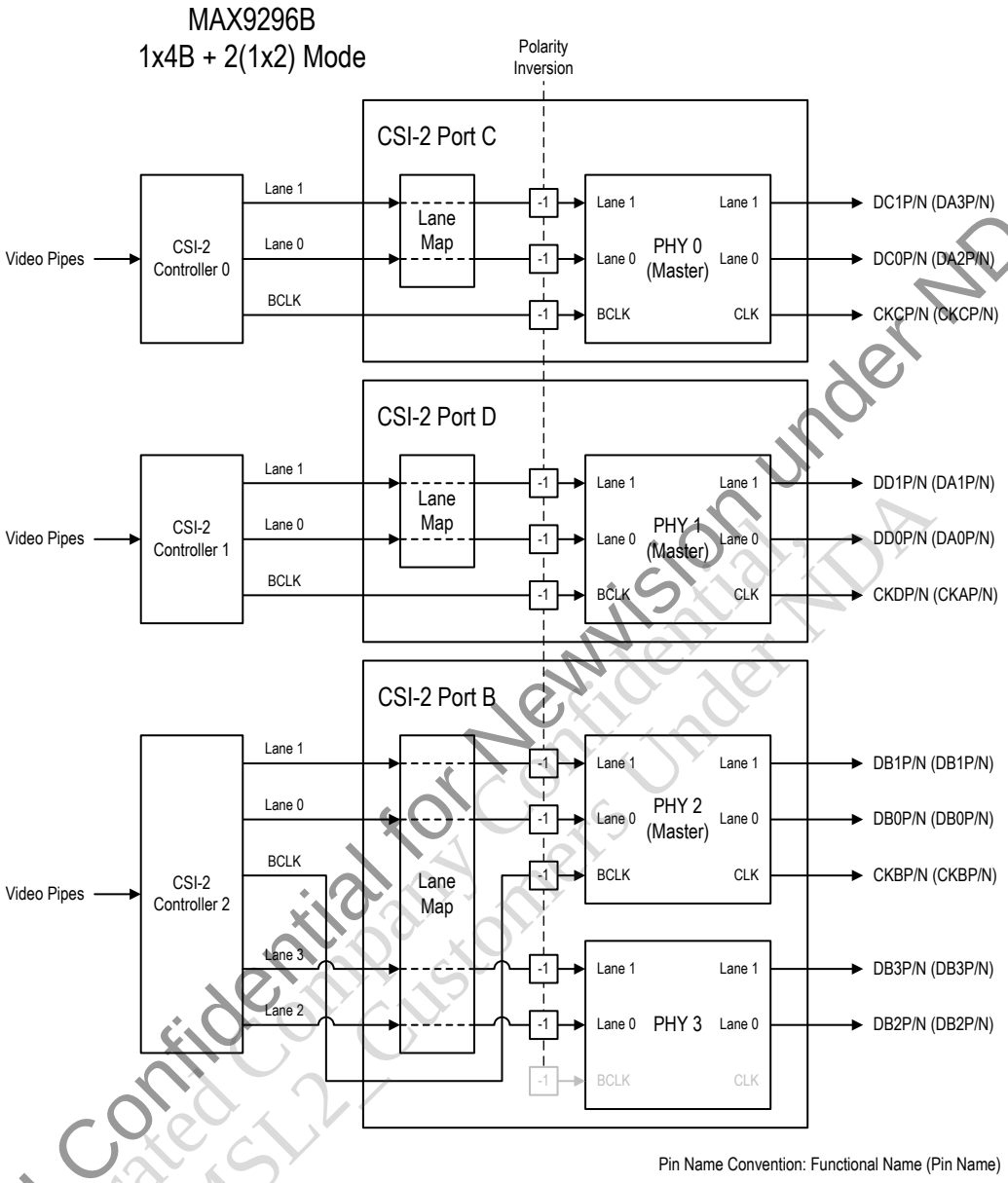


Figure 203. MAX9296B Deserializer D-PHY Rx – 1x4B + 2(1x2) Mode

ADI Confidential for Newvision under NDA
Analog Devices Integrated Company Confidential Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

43.4.4 Deserializer Lane Map/Swap and Polarity Inversion

Lane map and polarity inversion operates in the same manner as the serializer (see Serializer Lane Map/Swap and Polarity Inversion). Reference the figures in the CSI-2 Deserializer Port Architecture section to determine the required lane mapping for a specific device and application.

43.4.4.1 Lane Swap Example

Lane swap is illustrated in Figure 204 with a MAX9296B device in 1x4B + 2(1x2) mode. The register writes required for the depicted lane mapping are presented in Table 225 (reference the bolded decode entries).

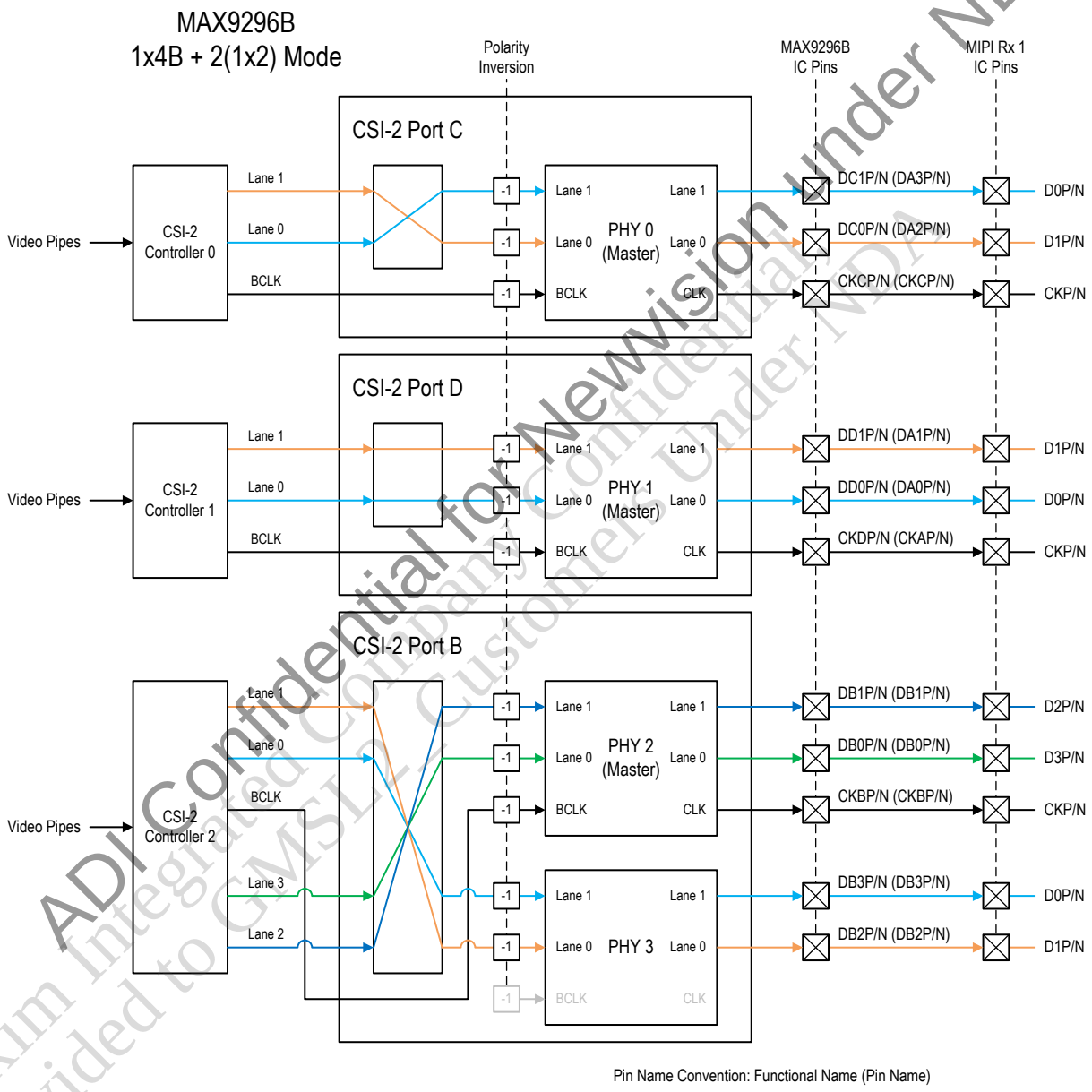


Figure 204. MAX9296B Deserializer – Lane Swap

ADI Confidential for Newvision Under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customer Under NDA

Table 225. MAX9296B Deserializer Lane Swap Example Register Writes

Register Address	Bitfield Name	Bits	POR	Decode
0x333	phy0_lane_map	3:0	MAX9296A: 0b0100 MAX9296B: 0b0100	0bXX00: PHY0 lane 0 is MIPI Controller lane 0 **0bXX01: PHY0 lane 0 is MIPI Controller lane 1** 0bXX10: PHY0 lane 0 is MIPI Controller lane 2 0bXX11: PHY0 lane 0 is MIPI Controller lane 3 **0b00XX: PHY0 lane 1 is MIPI Controller lane 0** 0b01XX: PHY0 lane 1 is MIPI Controller lane 1 0b10XX: PHY0 lane 1 is MIPI Controller lane 2 0b11XX: PHY0 lane 1 is MIPI Controller lane 3
0x333	phy1_lane_map	7:4	MAX9296A: 0b1110 MAX9296B: 0b0100	**0bXX00: PHY1 lane 0 is MIPI Controller lane 0** 0bXX01: PHY1 lane 0 is MIPI Controller lane 1 0bXX10: PHY1 lane 0 is MIPI Controller lane 2 0bXX11: PHY1 lane 0 is MIPI Controller lane 3 0b00XX: PHY1 lane 1 is MIPI Controller lane 0 **0b01XX: PHY1 lane 1 is MIPI Controller lane 1** 0b10XX: PHY1 lane 1 is MIPI Controller lane 2 0b11XX: PHY1 lane 1 is MIPI Controller lane 3
0x334	phy2_lane_map	3:0	MAX9296A: 0b0100 MAX9296B: 0b0100	0bXX00: PHY2 lane 0 is MIPI Controller lane 0 0bXX01: PHY2 lane 0 is MIPI Controller lane 1 0bXX10: PHY2 lane 0 is MIPI Controller lane 2 **0bXX11: PHY2 lane 0 is MIPI Controller lane 3** 0b00XX: PHY2 lane 1 is MIPI Controller lane 0 0b01XX: PHY2 lane 1 is MIPI Controller lane 1 **0b10XX: PHY2 lane 1 is MIPI Controller lane 2** 0b11XX: PHY2 lane 1 is MIPI Controller lane 3
0x334	phy3_lane_map	7:4	MAX9296A: 0b1110 MAX9296B: 0b0100	0bXX00: PHY3 lane 0 is MIPI Controller lane 0 **0bXX01: PHY3 lane 0 is MIPI Controller lane 1** 0bXX10: PHY3 lane 0 is MIPI Controller lane 2 0bXX11: PHY3 lane 0 is MIPI Controller lane 3 **0b00XX: PHY3 lane 1 is MIPI Controller lane 0** 0b01XX: PHY3 lane 1 is MIPI Controller lane 1 0b10XX: PHY3 lane 1 is MIPI Controller lane 2 0b11XX: PHY3 lane 1 is MIPI Controller lane 3

ADI Confidential for New Version Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customer_Under_NDA

43.4.4.1.1 Lane Swap Example Programming Summary

The following is a summary of the programming used in Figure 204:

- 0x333 = 0x41
  - phy0_lane_map = b0001
  - phy1_lane_map = b0100 (default, no swap)
- 0x334 = 0x1E
  - phy2_lane_map = b1011
  - phy3_lane_map = b0001

43.4.5 D-PHY Deskew

The MIPI D-PHY requires deskew calibration when operating above 1.5Gbps/lane. Each data lane supports clock-to-data lane deskew calibration. However, deskew is not programmed by the CSI-2 Configuration tool (GUI). See the

MIPI D-PHY Deskew section for configuration details.

43.4.6 Data Replication

GMSL2 CSI-2 deserializers have two methods of data replication:

1. The PHY copy function, and
2. Configuring video pipes to have duplicate stream IDs.

The PHY copy function takes the output from a specified source CSI-2 controller and routes it to the input of another destination CSI-2 controller (Figure 205). The data sent to the destination controller is already in CSI-2 format (as opposed to the generic GMSL2 format coming from the video pipes). This means both the source and destination controller must be configured for the same number of lanes and output data rate. When using PHY copy, the device must be configured so that the destination controller does not have any data routed to it from the video pipes. The two outputs (source and destination) will be exact copies.

There are two PHY copies available. The same source controller can be used for both PHY copies, but each destination controller must be unique. To use the PHY copy function, set the source controller using the phy_cp{0,1}_src bitfield (b00 = Controller 0, b01 = Controller 1, etc). Set the destination controller using the phy_cp{0,1}_dst bitfield. Enable the copy being used by setting the phy_cp0 and/or the phy_cp1 bit.

ADI Confidential Do not distribute or duplicate
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Confidential, Distribute only under NDA
Provided to GMSL2 Customers under NDA

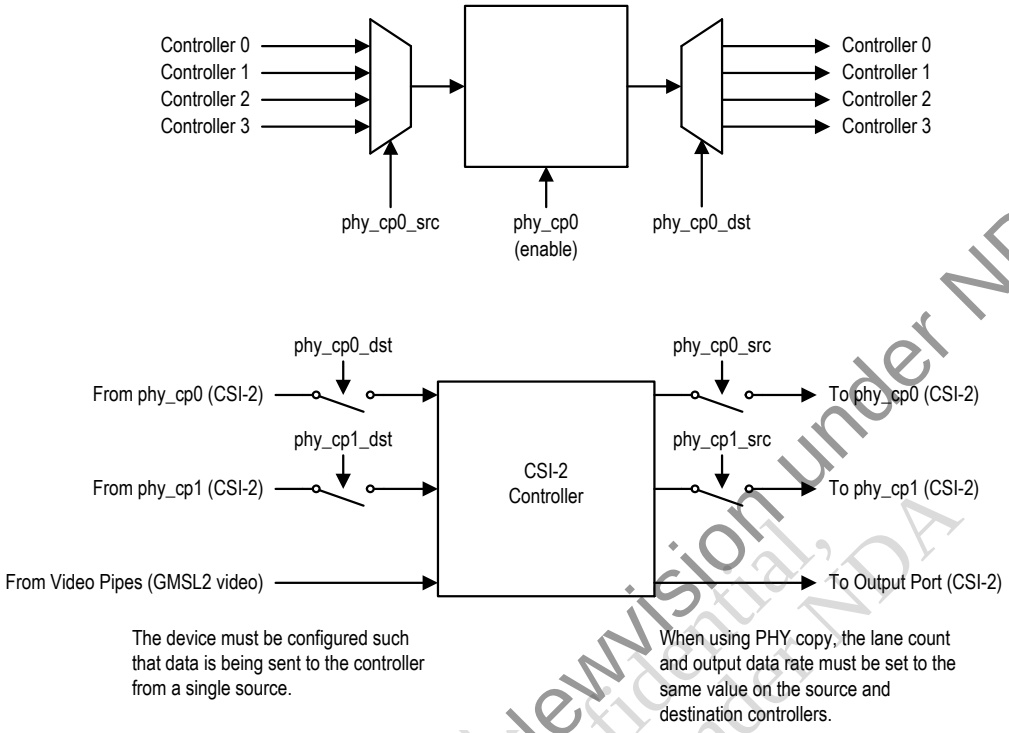


Figure 205. Data Replication Using the PHY Copy Function

Data replication via video pipes is used when data needs to be replicated to be transmitted by more than one MIPI output port with different configurations (i.e., the ports do not share the same number of data lanes or there is data present on one port that is not on the other). Data is replicated by assigning the same stream ID filter to multiple deserializer video pipes. For example, if a serializer has TX_STR_SEL = b00 set for pipe X and the deserializer has STR_SEL = b00 set for both pipe X and pipe Z, then the video data present in the serializer's pipe X will be duplicated in both pipe X and pipe Z in the deserializer. Data can then be mapped to two different controllers with completely independent pipe-to-controller routing and output configurations.

ADI Confidential for Newvision under NDA
Analog Devices Confidential,
Maxim Integrated Confidential,
Provided to GMSL2_CBB Users Under NDA

43.4.7 Status, Error, and Debug Bits

Table 226 lists the deserializer status and error bits. These bits do not assert ERRB but are critical when debugging a GMSL2 CSI-2 system. Note that these are generic video pipe errors and common to all GMSL2 devices.

Table 226. GMSL2 CSI-2 Deserializer Status and Error Registers

Register Address	Bitfield Name	Function	Decode
0x108 (Pipe X) **0x11A (Pipe Y)** **0x12C (Pipe Z)** **0x13E (Pipe U)**	VID_SEQ_ERR	Video Rx sequence error happened	Packets received out of expected order. This should not assert under normal conditions. 0b0: No error detected 0b1: Error detected
	VID_PKT_DET	Sufficient video Rx packet throughput detected	This should assert under normal conditions when video is locked. 0b0: Not enough throughput 0b1: Sufficient throughput detected
	VID_LOCK	Video pipeline locked.	Indicates video has been received by the deserializer and is present in this pipe. Under normal conditions, this should be continuously asserted. 0b0: Video pipeline not locked 0b1: Video pipeline locked
	VID_BLK_LEN_ERROR	Video Rx block length error detected	The received video data is not the length that is expected (as indicated by the metadata sent from the serializer). 0b0: No error detected 0b1: Error detected
0x312	LMO_{X,Y,Z,U}	Line Memory Overflow	Each bit indicates a video data overflow for the respective video pipe. This commonly occurs when the MIPI output rate is too slow for the incoming video throughput or the horizontal line length is too large. Can also assert if all of the data in the pipe is not routed to a controller (if VC1 is present in the pipe, but not routed to a controller, it may accumulate in the FIFO).
	cmd_overflow{1,2,3,4}	Command FIFO Overflow	Each bit indicates a command data overflow for the respective video pipe. This commonly occurs when the MIPI output rate is too slow for the incoming video throughput or the horizontal line length is too large. Can also assert if all of the data in the pipe is not routed to a controller (if pixel data is routed, but not VS or HS, this error may assert).
0x313	BACKTOP_MEM_CRC_ERR	Backtop memory CRC error flag	0b0: No CRC error occurred 0b1: CRC error occurred
0x540	VS_OUT1	Output VSYNC from VS1 pin.	MSB is enable. 2 LSBs select which pipe sources the VS output. 0b100: output pipe X 0b101: output pipe Y 0b110: output pipe Z 0b111: output pipe U
0x541	VS_OUT2	Output VSYNC from VS2 pin.	MSB is enable. 2 LSBs select which pipe sources the VS output. 0b100: output pipe X 0b101: output pipe Y 0b110: output pipe Z 0b111: output pipe U
0x542	DE_OUT1	Output DE from DE1 pin.	MSB is enable. 2 LSBs select which pipe sources the DE output. 0b100: output pipe X 0b101: output pipe Y 0b110: output pipe Z 0b111: output pipe U

ADI Confidential for New Division under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers under NDA

43.5 CSI-2 System Configuration

43.5.1 General Procedure

1. Determine the system information outlined in Required System Information.
2. Establish GMSL2 Link Lock (see GMSL2 Link Basics)
3. Serializer Configuration:
    a. D-PHY Rx Configuration
    b. Serializer Lane Map/Swap and Polarity Inversion
    c. D-PHY Deskew (if input MIPI data rate is >1500Mbps/lane)
    d. Video Pipes and GMSL Core
4. Deserializer Configuration:
    a. GMSL Core and Video Pipes
    b. Pipe-to-Controller Mapping
    c. D-PHY Tx Configuration
    d. Deserializer Lane Map/Swap and Polarity Inversion
    e. D-PHY Deskew (if output MIPI data rate is >1500Mbps/lane)
    f. Data Replication
5. Enable the MIPI Source

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

43.5.2 Required System Information

Before configuring the GMSL2 CSI-2 serial link system, define the following use case attributes:

43.5.2.1 CSI-2 Serializer/CSI Source

1. Video Source Information
    a. Total resolution (active pixels + blanking + embedded data)
    b. FPS
    c. BPP and data types (see Table 208)
    d. Number of VCs and their interleaving (see VS Routing)
2. MIPI Port Definition
    a. Number of ports
    b. Number of data lanes per port
    c. MIPI bit rate/clock of each port
    d. Video carried on each port (see step 1)
3. PCB Layout Information
    a. Lane swapping
    b. Polarity inversion

43.5.2.2 CSI-2 Deserializer/CSI Sink

1. Number of serializers and the data coming from each (defined above)
2. MIPI Port Definition
    a. Number of ports
    b. Number of data lanes per port
    c. MIPI bit rate/clock of each port
    d. Video carried on each port
3. PCB Layout Information
    a. Lane swapping
    b. Polarity inversion

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.5.3 CSI-2 Configuration Tool

ADI provides software to automate the configuration of GMSL2 CSI-2 serial link systems. *Table 227* contains a list of use cases supported by the GMSL2 GUI.

Table 227. Use Cases Supported by the CSI-2 Configuration Tool

Single/Dual Serializer	Number of Pixel Datatypes	Embedded Data	VC Line Interleaving	VC Frame Interleaving
Single	1	No	No	No
Single	1	No	Yes	No
Single	1	Yes	No	No
Single	1	Yes	Yes	No
Dual	1	No	No	No

43.5.4 Simple Use Case Example

Simple use cases (e.g., single imager, single data type, minimal bandwidth requirements) can be brought up with very few register writes by utilizing the default video routings shown in Table 228 and Table 229. For maximum bandwidth utilization and/or more complicated use cases, additional configuration is required. Use the CSI-2 Configuration software tool if the use case is listed in Table 227.

Table 228. GMSL2 CSI-2 Serializer Default Routing

Part Number / Mode	Input Port	CSI-2 PHYs	CSI-2 Controller	Video Pipe	Stream ID
MAX9295A/B 1x4 Mode	Port B	PHY1, PHY2 (master)	Controller 1	Pipe Z	2
MAX9295B 2x2 Mode	Port A	PHY1 (master)	Controller 0	Pipe X	0
	Port B	PHY2 (master)	Controller 1	Pipe Z	2
MAX9295D/E 2x4 mode	Port A	PHY0, PHY1 (master)	Controller 0	Pipe X	0
	Port B	PHY2 (master), PHY3	Controller 1	Pipe Z	2

Table 229. GMSL2 CSI-2 Deserializer Default Routing

Part Number/Mode	Stream ID	Video Pipe	CSI-2 Controller	CSI-2 PHYs	Output Port
MAX9296A/B 2x4 Mode	1	Pipe Y	Controller 1	PHY0, PHY1 (master)	Port A
	2	Pipe Z	Controller 2	PHY2 (master), PHY3	Port B
MAX9296B 4x2 Mode	0	Pipe X	Controller 0	PHY0 (master)	Port C
	1	Pipe Y	Controller 1	PHY1 (master)	Port D
	2	Pipe Z	Controller 2	PHY2 (master)	Port E
	3	Pipe U	Controller 3	PHY3 (master)	Port F

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

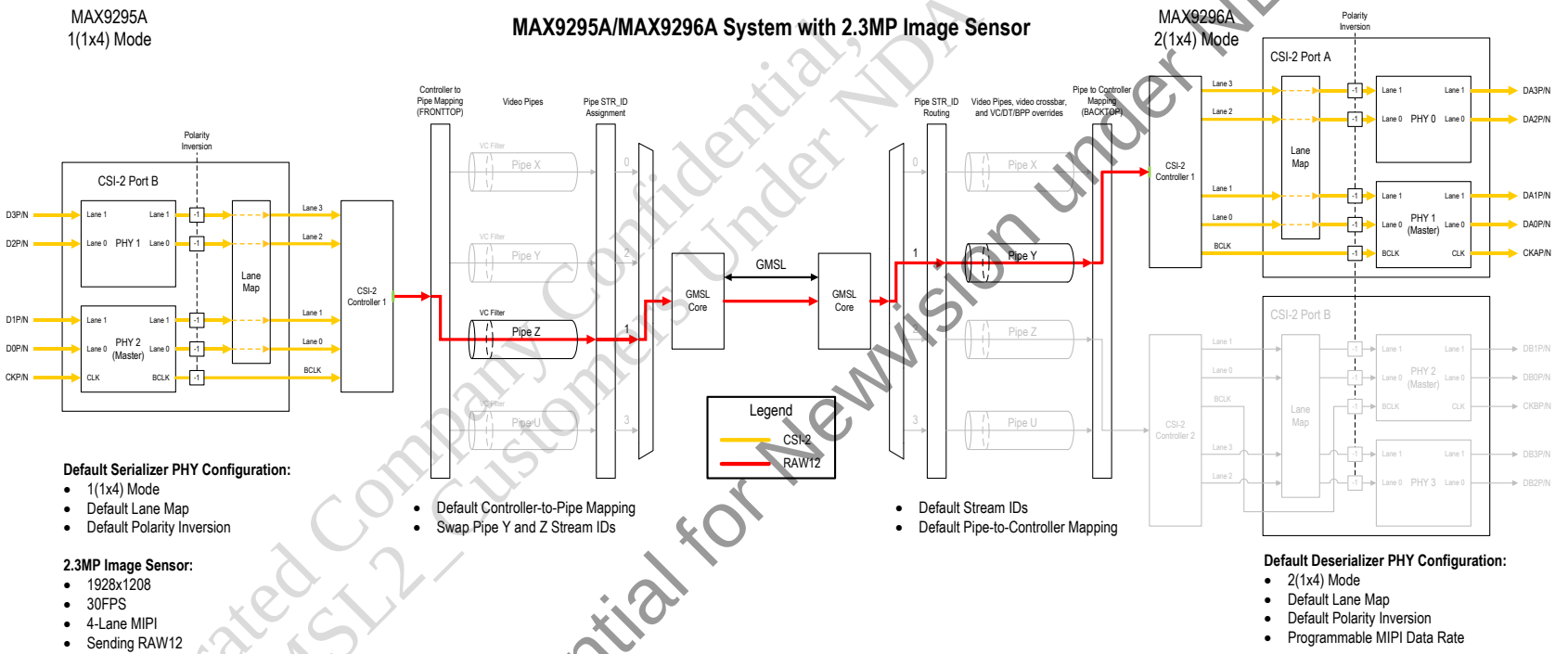


Figure 206. MAX9295A/MAX9296A Using 2.3MP Image Sensor (Simple Use Case)

Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
ADI Confidential for Newvision under NDA

For the simple use case presented in Figure 206, a single data type is input to the MIPI port on the MAX9295A serializer and is automatically sent to Controller 1. The default controller-to-pipe mapping routes the output of Controller 1 to pipe Z. To utilize the default pipe-to-controller mapping on the deserializer, the stream ID in the serializer is configured so that the video data in serializer pipe Z is routed to deserializer pipe Y. Data in deserializer pipe Y is output on Port A by default. To do this, the stream ID for pipe Y and pipe Z are swapped in the serializer. The programming sequence is provided below.

43.5.4.1 Programming Sequence (Simple Use Case)

1. Establish GMSL2 Lock (see GMSL2 Link Basics)
2. Serializer Configuration:
    a. MIPI D-PHY Rx (default)
        i. Port configuration: 1x4
        ii. Lane count: 4
        iii. Lane map (default)
        iv. Polarity inversion (default)
    b. Controller-to-Pipe mapping (default)
    c. Pipe Configuration
        i. Pipe Stream IDs: pipe Y = 2, pipe Z = 1
        ii. Pipe Tx Enable (default)
3. Deserializer Configuration:
    a. Pipe Stream IDs (default)
    b. Pipe-to-Controller mapping (default)
    c. MIPI D-PHY Tx
        i. Port configuration: 2x4 (default)
        ii. Lane count: 4-lane (default)
        iii. Lane map (default)
        iv. Polarity inversion (default)
        v. MIPI data rate: 800mbps/lane
4. Enable the MIPI Source

43.5.4.2 Summary of Register Writes (Simple Use Case)

1. MAX9295A (Serializer)
    a. Set pipe Y stream ID to 2: ADDR 0x57 = 0x12
    b. Set pipe Z stream ID to 1: ADDR 0x5B = 0x11
2. MAX9296A (Deserializer)
    a. Set Port A/PHY1 bit rate to 800mbps/lane: ADDR 0x320 = 0x28

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

43.5.5 Advanced Use Case Example

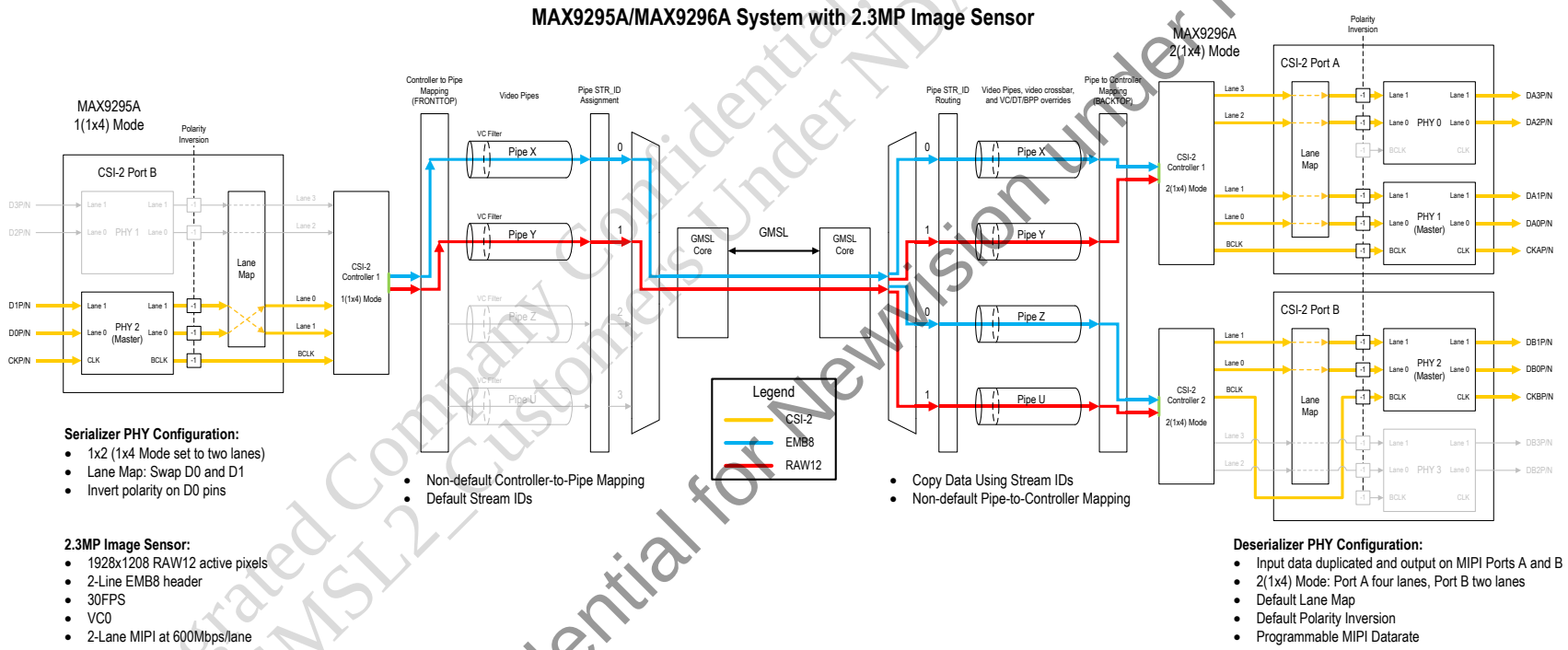
Note: This use case is performed using the CSI-2 Configuration Tool in the GMSL2 GUI.

An advanced use case using MAX9295A/MAX9296A is presented in Figure 207. This application includes an image sensor outputting 1928x1200 30fps RAW12 frames on VC0. The sensor is configured to output CSI-2 data on two lanes at 600Mbps/lane. Within each frame is an EMB8 header that is 2 lines long. Lane swapping and polarity inversion are used to optimize the camera routing.

On the ECU side, the imager data must be duplicated and sent to two different SoCs. The two SoC CSI-2 ports must operate with a different number of lanes. Data must be duplicated with video pipe duplication (the PHY copy function cannot be used). The input data must be contained within a maximum of two pipes so that video pipes are available for data duplication.

This use case is supported by the CSI-2 Configuration Tool. The output is shown below.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA



Serializer PHY Configuration:

- 1x2 (1x4 Mode set to two lanes)
- Lane Map: Swap D0 and D1
- Invert polarity on D0 pins

2.3MP Image Sensor:

- 1928x1208 RAW12 active pixels
- 2-Line EMB8 header
- 30FPS
- VC0
- 2-Lane MIPI at 600Mbps/lane

- Non-default Controller-to-Pipe Mapping
- Default Stream IDs

- Copy Data Using Stream IDs
- Non-default Pipe-to-Controller Mapping

Deserializer PHY Configuration:

- Input data duplicated and output on MIPI Ports A and B
- 2(1x4) Mode: Port A four lanes, Port B two lanes
- Default Lane Map
- Default Polarity Inversion
- Programmable MIPI Datarate

Figure 207. MAX9295A/MAX9296A Using 2.3MP Image Sensor (Advanced Use Case)

Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA
ADI Confidential for NewVision under NDA

43.5.5.1 Programming Sequence (Advanced Use Case)

1. Establish GMSL2 Lock (see Link Basics section)
2. Serializer Configuration:
    a. MIPI D-PHY Rx
        i. Port configuration: 1x4 (default)
        ii. Lane count: 2-lanes
        iii. Lane map: swap D0 and D1
        iv. Polarity inversion: invert polarity on D0 (serializer IC pins)
    b. Controller-to-Pipe mapping
        i. Data type filtering: route EMB8 to pipe X, RAW12 to pipe Y
        ii. Pipe-to-Controller Select: Controller 1/MIPI Port B to pipe X and Y
    c. Pipe Configuration
        i. Pipe Stream IDs (default)
        ii. Pipe Tx Enable: Enable TX on pipe X and pipe Y
    d. Other
        i. Enable double mode for pipe X and pipe Y (maximizes BW efficiency)
3. Deserializer Configuration:
    a. Pipe Stream IDs
        i. Copy data by duplicating Stream IDs
    b. Pipe-to-Controller mapping
        i. Route pipe X and pipe Y to Controller 1 (MIPI Port A)
        ii. Route pipe Z and pipe U to Controller 2 (MIPI Port B)
        iii. Disable VSYNC in pipe Y and pipe Z
    c. MIPI D-PHY Tx
        i. Port configuration: 2x4 (default)
        ii. Lane count: 4-lane on Port A (default), 2-lane on Port B
        iii. Lane map (default)
        iv. Polarity inversion (default)
        v. MIPI data rate
            1. Port A = 1200Mbps/lane
            2. Port B = 1500Mbps/lane
4. Enable the MIPI Source

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.5.5.2 CSI-2 Configuration Tool Output

```
//////////////////////////////////////////////////////
// THIS DATA FILE, AND ALL INFORMATION CONTAINED THEREIN, IS PROVIDED "AS IS",
// WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
// THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
// IN NO EVENT SHALL MAXIM INTEGRATED BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
// WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
// CONNECTION WITH THE DATA FILE, THE INFORMATION CONTAINED THEREIN, OR ITS USE FOR ANY PURPOSE.
// BEFORE USING THIS DATA FILE IN ANY APPLICATION FOR PRODUCTION OR DEPLOYMENT, THE CUSTOMER
// IS SOLELY RESPONSIBLE FOR TESTING AND VERIFYING THE CONTENT OF THIS DATA FILE IN CONNECTION
// WITH THEIR PRODUCTS AND SYSTEM(S).
//////////////////////////////////////////////////////
//
// Script is generated by GMSL CSI Configuration Tool Version 0.6.1
//
//
//////////////////////////////////////////////////////
// Instructions for Serializer 1
//////////////////////////////////////////////////////
0x04, 0x80, 0x00, 0x10, 0x21, // Apply Reset Oneshot for changes
0x00, 0x80,
0x04, 0x80, 0x01, 0x02, 0x0C, // Disable heart beat and disable drift detect for pipe X
0x04, 0x80, 0x01, 0x0A, 0x0C, // Disable heart beat and disable drift detect for pipe Y
0x04, 0x80, 0x01, 0x12, 0x0C, // Disable heart beat and disable drift detect for pipe Z
0x04, 0x80, 0x01, 0x1A, 0x0C, // Disable heart beat and disable drift detect for pipe U
0x04, 0x80, 0x03, 0x30, 0x00, // Port Configuration
0x04, 0x80, 0x03, 0x31, 0x13, // Port set lane count
0x04, 0x80, 0x03, 0x32, 0xEE, // Port lane swapping 2 Port lane swapping 3
0x04, 0x80, 0x03, 0x33, 0xE1, // Port lane swapping 0 Port lane swapping 1
0x04, 0x80, 0x03, 0x34, 0x00, // Port polarity 2 Port polarity 3
0x04, 0x80, 0x03, 0x35, 0x01, // Port polarity 0 Port polarity 1 Port polarity clock
0x04, 0x80, 0x03, 0x11, 0x35, // Start Port Pipe X Start Port Pipe Y
0x04, 0x80, 0x03, 0x08, 0x7F, // Pipe X clock selected Pipe Y clock selected Pipe Z clock selected Pipe U clock
selected
0x04, 0x80, 0x00, 0x02, 0xF3, // Pipe X video transmission enable Pipe Y video transmission enable Pipe Z video
transmission enable Pipe U video transmission enable
0x04, 0x80, 0x00, 0x53, 0x10, // Pipe X stream select 0
0x04, 0x80, 0x00, 0x57, 0x11, // Pipe Y stream select 1
0x04, 0x80, 0x00, 0x5B, 0x12, // Pipe Z stream select 2
0x04, 0x80, 0x00, 0x5F, 0x13, // Pipe U stream select 3
0x04, 0x80, 0x01, 0x00, 0x68, //
0x04, 0x80, 0x01, 0x10, 0x68, //
0x04, 0x80, 0x01, 0x01, 0x58, //
0x04, 0x80, 0x01, 0x11, 0x58, //
0x04, 0x80, 0x03, 0x12, 0x01, // BPP 8 Double
0x04, 0x80, 0x03, 0x13, 0x20, // BPP 10 Double BPP 12 Double
```

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential, Provided to Customers Under NDA

```
0x04, 0x80, 0x03, 0x14, 0x52, // Memory data type 1 Pipe X
0x04, 0x80, 0x03, 0x15, 0x00, // Memory data type 2 Pipe X
0x04, 0x80, 0x03, 0x16, 0x6C, // Memory data type 1 Pipe Y
0x04, 0x80, 0x03, 0x17, 0x00, // Memory data type 2 Pipe Y
0x04, 0x80, 0x03, 0x18, 0x00, // Memory data type 1 Pipe Z
0x04, 0x80, 0x03, 0x19, 0x00, // Memory data type 2 Pipe Z
0x04, 0x80, 0x03, 0x1A, 0x00, // Memory data type 1 Pipe U
0x04, 0x80, 0x03, 0x1B, 0x00, // Memory data type 2 Pipe U
0x04, 0x80, 0x03, 0x1C, 0x30, // Soft BPP Pipe X
0x04, 0x80, 0x03, 0x1D, 0x38, // Soft BPP Pipe Y
0x04, 0x80, 0x03, 0x1E, 0x00, // Soft BPP Pipe Z
0x04, 0x80, 0x03, 0x1F, 0x00, // Soft BPP Pipe U
//
////////////////////////////////////////////////////////
// Instructions for Deserializer MAX9296A
////////////////////////////////////////////////////////
0x04, 0x90, 0x00, 0x10, 0x11, // Auto Link for 1 serializer Apply reset oneshot for changes
0x00, 0x80,
0x04, 0x90, 0x03, 0x20, 0x2C, // Port A clock speed
0x04, 0x90, 0x03, 0x23, 0x2F, // Port B clock speed
0x04, 0x90, 0x01, 0x00, 0x23, // Seq_miss_en fixed to 0 for CSI Dis_pkt_det fixed to 1 for CSI
0x04, 0x90, 0x01, 0x12, 0x23, // Seq_miss_en fixed to 0 for CSI Dis_pkt_det fixed to 1 for CSI
0x04, 0x90, 0x01, 0x24, 0x23, // Seq_miss_en fixed to 0 for CSI Dis_pkt_det fixed to 1 for CSI
0x04, 0x90, 0x01, 0x36, 0x23, // Seq_miss_en fixed to 0 for CSI Dis_pkt_det fixed to 1 for CSI
0x04, 0x90, 0x03, 0x30, 0x04, // Set Deserializer to 2x4 Mode
0x04, 0x90, 0x03, 0x32, 0xF0, //
0x04, 0x90, 0x03, 0x33, 0x4E, // Port A lane swapping 0 Port A lane swapping 1 Port A lane swapping 2 Port A
lane swapping 3
0x04, 0x90, 0x03, 0x34, 0xE4, // Port B lane swapping 0 Port B lane swapping 1 Port B lane swapping 2 Port B
lane swapping 3
0x04, 0x90, 0x04, 0x0A, 0x10, //
0x04, 0x90, 0x04, 0x4A, 0xD0, //
0x04, 0x90, 0x04, 0x8A, 0x50, //
0x04, 0x90, 0x04, 0xCA, 0x10, //
0x04, 0x90, 0x03, 0x26, 0x2F, //
0x04, 0x90, 0x00, 0x50, 0x00, // Pipe X stream select 0
0x04, 0x90, 0x00, 0x51, 0x01, // Pipe Y stream select 1
0x04, 0x90, 0x00, 0x52, 0x00, // Pipe Z stream select 2
0x04, 0x90, 0x00, 0x53, 0x01, // Pipe U stream select 3
0x04, 0x90, 0x03, 0x35, 0x00, // Port A polarity 0 Port A polarity 1 Port A polarity 2 Port A polarity 3 Port A
polarity clock
0x04, 0x90, 0x03, 0x36, 0x00, // Port B polarity 0 Port B polarity 1 Port B polarity 2 Port B polarity 3 Port B
polarity clock
0x04, 0x90, 0x03, 0x39, 0x00, //
0x04, 0x90, 0x03, 0x3A, 0x02, //
0x04, 0x90, 0x03, 0x3B, 0x00, //
0x04, 0x90, 0x03, 0x3C, 0x02, //
0x04, 0x90, 0x04, 0x33, 0x00, //
```

ADI Confidential for New Vision under NDA
Material Confidential, Distribute only under NDA
Provided to ADI_2_ Customers Under NDA
Material Confidential, Company Confidential,

```
0x04, 0x90, 0x04, 0x73, 0x03, // Un-double data CTRL1
0x04, 0x90, 0x04, 0xB3, 0x03, // Un-double data CTRL2
0x04, 0x90, 0x04, 0xF3, 0x00, //
0x04, 0x90, 0x04, 0x0B, 0x07, // Enable mappings for Pipe X
0x04, 0x90, 0x04, 0x0C, 0x00, // Enable mappings for Pipe X
0x04, 0x90, 0x04, 0x4B, 0x01, // Enable mappings for Pipe Y
0x04, 0x90, 0x04, 0x4C, 0x00, // Enable mappings for Pipe Y
0x04, 0x90, 0x04, 0x8B, 0x07, // Enable mappings for Pipe Z
0x04, 0x90, 0x04, 0x8C, 0x00, // Enable mappings for Pipe Z
0x04, 0x90, 0x04, 0xCB, 0x01, // Enable mappings for Pipe U
0x04, 0x90, 0x04, 0xCC, 0x00, // Enable mappings for Pipe U
0x04, 0x90, 0x04, 0x2D, 0x15, // Destination for Pipe X
0x04, 0x90, 0x04, 0x2E, 0x00, // Destination for Pipe X
0x04, 0x90, 0x04, 0x2F, 0x00, // Destination for Pipe X
0x04, 0x90, 0x04, 0x30, 0x00, // Destination for Pipe X
0x04, 0x90, 0x04, 0x6D, 0x15, // Destination for Pipe Y
0x04, 0x90, 0x04, 0x6E, 0x00, // Destination for Pipe Y
0x04, 0x90, 0x04, 0x6F, 0x00, // Destination for Pipe Y
0x04, 0x90, 0x04, 0x70, 0x00, // Destination for Pipe Y
0x04, 0x90, 0x04, 0xAD, 0x2A, // Destination for Pipe Z
0x04, 0x90, 0x04, 0xAE, 0x00, // Destination for Pipe Z
0x04, 0x90, 0x04, 0xAF, 0x00, // Destination for Pipe Z
0x04, 0x90, 0x04, 0xB0, 0x00, // Destination for Pipe Z
0x04, 0x90, 0x04, 0xED, 0x2A, // Destination for Pipe U
0x04, 0x90, 0x04, 0xEE, 0x00, // Destination for Pipe U
0x04, 0x90, 0x04, 0xEF, 0x00, // Destination for Pipe U
0x04, 0x90, 0x04, 0xF0, 0x00, // Destination for Pipe U
0x04, 0x90, 0x04, 0x0D, 0x12, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x0E, 0x12, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x0F, 0x00, // Pipe X Frame Start Source
0x04, 0x90, 0x04, 0x10, 0x00, // Pipe X Frame Start Destination
0x04, 0x90, 0x04, 0x11, 0x01, // Pipe X Frame End Source
0x04, 0x90, 0x04, 0x12, 0x01, // Pipe X Frame End Destination
0x04, 0x90, 0x04, 0x4D, 0x2C, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x4E, 0x2C, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x4F, 0x00, // Pipe Y Frame Start Source
0x04, 0x90, 0x04, 0x50, 0x00, // Pipe Y Frame Start Destination
0x04, 0x90, 0x04, 0x51, 0x00, // Pipe Y Frame End Source
0x04, 0x90, 0x04, 0x52, 0x00, // Pipe Y Frame End Destination
0x04, 0x90, 0x04, 0x8D, 0x12, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x8E, 0x12, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x8F, 0x00, // Pipe Z Frame Start Source
0x04, 0x90, 0x04, 0x90, 0x00, // Pipe Z Frame Start Destination
0x04, 0x90, 0x04, 0x91, 0x01, // Pipe Z Frame End Source
0x04, 0x90, 0x04, 0x92, 0x01, // Pipe Z Frame End Destination
0x04, 0x90, 0x04, 0xCD, 0x2C, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xCE, 0x2C, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xCF, 0x00, // Pipe U Frame Start Source
```

Analog Confidential for NewVision under NDA
Material Intended for Distribution only under NDA
Protected Content per ANALOG IP Policy
Analog Confidential, Distribute to Customers Under NDA

```
0x04, 0x90, 0x04, 0xD0, 0x00, // Pipe U Frame Start Destination
0x04, 0x90, 0x04, 0xD1, 0x00, // Pipe U frame End Source
0x04, 0x90, 0x04, 0xD2, 0x00, // Pipe U frame End Destination
0x04, 0x90, 0x03, 0x09, 0x01, // Disabled VC VSYNC Check for pipe X (VC 0-7)
0x04, 0x90, 0x03, 0x0B, 0x01, // Disabled VC VSYNC Check for pipe Y (VC 0-7)
0x04, 0x90, 0x03, 0x0D, 0x01, // Disabled VC VSYNC Check for pipe Z (VC 0-7)
0x04, 0x90, 0x03, 0x0F, 0x01, // Disabled VC VSYNC Check for pipe U (VC 0-7)
0x04, 0x90, 0x04, 0x13, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x14, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x15, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x16, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x17, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x18, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x19, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x1A, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x1B, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x1C, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x1D, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x1E, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x1F, 0x00, // Pipe X DT and VC Source
0x04, 0x90, 0x04, 0x20, 0x00, // Pipe X DT and VC Destination
0x04, 0x90, 0x04, 0x21, 0x00, // Pipe X Frame Start Source
0x04, 0x90, 0x04, 0x22, 0x00, // Pipe X Frame Start Destination
0x04, 0x90, 0x04, 0x23, 0x00, // Pipe X Frame End Source
0x04, 0x90, 0x04, 0x24, 0x00, // Pipe X Frame End Destination
0x04, 0x90, 0x04, 0x25, 0x00, // Pipe X Frame Start Source
0x04, 0x90, 0x04, 0x26, 0x00, // Pipe X Frame Start Destination
0x04, 0x90, 0x04, 0x27, 0x00, // Pipe X Frame End Source
0x04, 0x90, 0x04, 0x28, 0x00, // Pipe X Frame End Destination
0x04, 0x90, 0x04, 0x29, 0x00, // Pipe X Frame Start Source
0x04, 0x90, 0x04, 0x2A, 0x00, // Pipe X Frame Start Destination
0x04, 0x90, 0x04, 0x2B, 0x00, // Pipe X Frame End Source
0x04, 0x90, 0x04, 0x2C, 0x00, // Pipe X Frame End Destination
0x04, 0x90, 0x04, 0x53, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x54, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x55, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x56, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x57, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x58, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x59, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x5A, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x5B, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x5C, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x5D, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x5E, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x5F, 0x00, // Pipe Y DT and VC Source
0x04, 0x90, 0x04, 0x60, 0x00, // Pipe Y DT and VC Destination
0x04, 0x90, 0x04, 0x61, 0x00, // Pipe Y Frame Start Source
```

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Distribute only under NDA
Proprietary to Maxim Integrated Products, Inc. Confidential, Proprietary to Customers Under NDA

0x04, 0x90, 0x04, 0x62, 0x00, // Pipe Y Frame Start Destination
0x04, 0x90, 0x04, 0x63, 0x00, // Pipe Y Frame End Source
0x04, 0x90, 0x04, 0x64, 0x00, // Pipe Y Frame End Destination
0x04, 0x90, 0x04, 0x65, 0x00, // Pipe Y Frame Start Source
0x04, 0x90, 0x04, 0x66, 0x00, // Pipe Y Frame Start Destination
0x04, 0x90, 0x04, 0x67, 0x00, // Pipe Y Frame End Source
0x04, 0x90, 0x04, 0x68, 0x00, // Pipe Y Frame End Destination
0x04, 0x90, 0x04, 0x69, 0x00, // Pipe Y Frame Start Source
0x04, 0x90, 0x04, 0x6A, 0x00, // Pipe Y Frame Start Destination
0x04, 0x90, 0x04, 0x6B, 0x00, // Pipe Y Frame End Source
0x04, 0x90, 0x04, 0x6C, 0x00, // Pipe Y Frame End Destination
0x04, 0x90, 0x04, 0x93, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x94, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x95, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x96, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x97, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x98, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x99, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x9A, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x9B, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x9C, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x9D, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0x9E, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0x9F, 0x00, // Pipe Z DT and VC Source
0x04, 0x90, 0x04, 0xA0, 0x00, // Pipe Z DT and VC Destination
0x04, 0x90, 0x04, 0xA1, 0x00, // Pipe Z Frame Start Source
0x04, 0x90, 0x04, 0xA2, 0x00, // Pipe Z Frame Start Destination
0x04, 0x90, 0x04, 0xA3, 0x00, // Pipe Z Frame End Source
0x04, 0x90, 0x04, 0xA4, 0x00, // Pipe Z Frame End Destination
0x04, 0x90, 0x04, 0xA5, 0x00, // Pipe Z Frame Start Source
0x04, 0x90, 0x04, 0xA6, 0x00, // Pipe Z Frame Start Destination
0x04, 0x90, 0x04, 0xA7, 0x00, // Pipe Z Frame End Source
0x04, 0x90, 0x04, 0xA8, 0x00, // Pipe Z Frame End Destination
0x04, 0x90, 0x04, 0xA9, 0x00, // Pipe Z Frame Start Source
0x04, 0x90, 0x04, 0xAA, 0x00, // Pipe Z Frame Start Destination
0x04, 0x90, 0x04, 0xAB, 0x00, // Pipe Z Frame End Source
0x04, 0x90, 0x04, 0xAC, 0x00, // Pipe Z Frame End Destination
0x04, 0x90, 0x04, 0xD3, 0x00, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xD4, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xD5, 0x00, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xD6, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xD7, 0x00, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xD8, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xD9, 0x00, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xDA, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xDB, 0x00, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xDC, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xDD, 0x00, // Pipe U DT and VC Source

Analog Confidential – Do Not Distribute or Duplicate
Maxim Integrated Confidential, Provided to Customers Under NDA
Analog Confidential – Distribute only under NDA
Maxim Integrated Confidential, Provided to Customers Under NDA

```
0x04, 0x90, 0x04, 0xDE, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xDF, 0x00, // Pipe U DT and VC Source
0x04, 0x90, 0x04, 0xE0, 0x00, // Pipe U DT and VC Destination
0x04, 0x90, 0x04, 0xE1, 0x00, // Pipe U Frame Start Source
0x04, 0x90, 0x04, 0xE2, 0x00, // Pipe U Frame Start Destination
0x04, 0x90, 0x04, 0xE3, 0x00, // Pipe U Frame End Source
0x04, 0x90, 0x04, 0xE4, 0x00, // Pipe U Frame End Destination
0x04, 0x90, 0x04, 0xE5, 0x00, // Pipe U Frame Start Source
0x04, 0x90, 0x04, 0xE6, 0x00, // Pipe U Frame Start Destination
0x04, 0x90, 0x04, 0xE7, 0x00, // Pipe U Frame End Source
0x04, 0x90, 0x04, 0xE8, 0x00, // Pipe U Frame End Destination
0x04, 0x90, 0x04, 0xE9, 0x00, // Pipe U Frame Start Source
0x04, 0x90, 0x04, 0xEA, 0x00, // Pipe U Frame Start Destination
0x04, 0x90, 0x04, 0xEB, 0x00, // Pipe U Frame End Source
0x04, 0x90, 0x04, 0xEC, 0x00, // Pipe U Frame End Destination
0x04, 0x90, 0x03, 0x13, 0x02, //
0x04, 0x90, 0x03, 0x14, 0x00, //
0x04, 0x90, 0x03, 0x15, 0x00, //
0x04, 0x90, 0x03, 0x16, 0x00, //
0x04, 0x90, 0x03, 0x17, 0x00, //
0x04, 0x90, 0x03, 0x1A, 0x00, //
0x04, 0x90, 0x03, 0x18, 0x00, //
0x04, 0x90, 0x03, 0x19, 0x00, //
0x04, 0x90, 0x03, 0x1D, 0x2F, //
0x04, 0x90, 0x03, 0x1C, 0x10, // Process BPP=8 as 16-bit color
0x04, 0x90, 0x03, 0x1F, 0x10, // Enable 8-bit write alternate map to RAM
0x04, 0x90, 0x01, 0xD9, 0x19, //
0x04, 0x90, 0x01, 0xF9, 0x39, // Remove VSYNC from Pipe Y
0x04, 0x90, 0x02, 0x19, 0x19, //
0x04, 0x90, 0x02, 0x39, 0x39, // Remove VSYNC from Pipe U
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

43.6 Video PRBS Testing

The forward channel video PRBS evaluates the video channels by testing the BER of video packets. Refer to the *PRBS Testing* section for details.

The bits per pixel value of CSI-2 video formats is usually different than 24bpp and it must be overridden to 24bpp with soft_bpp{x,y,z,u} and soft_bpp{x,y,z,u}_en before performing video PRBS testing. In addition, if heartbeat is disabled (i.e., LIM_HEART = 1 on serializer, and SEQ_MISS_EN = 0 and DIS_PKT_DET = 1 on deserializer in the used video pipes), after overriding bpp to 24 on the serializer, and before enabling video PRBS generator/checker, VID_EN_{X,Y,Z,U} needs to be toggled to 0 and then 1 again in the used video pipe(s) on the deserializer. Or heartbeat needs to be enabled before performing VPRBS testing. Otherwise VPRBS_FAIL will be asserted and VPRBS_ERR will be set to 0xFF in the used video pipe(s).

The following special test sequence is to configure a CSI-2 serializer and a CSI-2 deserializer to perform the VPRBS test while video is running.

1. Configure GMSL2 parts and the image sensor for normal operation
    a. Configure deserializer and serializer first, then image sensor
    b. Enable image sensor output
    c. Check PCLKDET =1 and VID_LOCK =1 in the used video pipes, the system is error free and working well
2. Perform video PRBS testing
    a. If bits per pixel value is different than 24bpp, apply the following steps. Otherwise, proceed with step 2-b.
        i. Stop image sensor output
        ii. Add as much delay as image sensor needs
        iii. Set AUTO_BPP to 1, if it is set to 0 for normal operation in step 1 above, and set soft_bpp{x,y,z,u} to 24 and soft_bpp{x,y,z,u}_en to 1 in the used video pipe(s) on serializer
        iv. If heartbeat is disabled for normal operation in step 1 above, apply one of the following steps. Otherwise, proceed with the next step.
            1. Toggle VID_EN_{X/Y/Z/U} register to 0 and then 1 again on deserializer
            2. Enable heartbeat both on serializer and deserializer for VPRBS testing
        v. Reenable image sensor output
        vi. Poll PCLKDET and VID_LOCK in the used video pipes. Before enabling video PRBS generator on serializer, be sure PCLKDET = 1 and VID_LOCK = 1 in the used video pipes.
    b. Set VID_PRBS_EN to 1 on serializer in the used video pipe(s)
    c. Set VPRBS_CHK_EN to 1 on deserializer in the used video pipe(s)
    d. Wait for a few msecs
    e. Read register 0x13 on deserializer to verify ERROR = 0 and LOCKED = 1
    f. Check for video lock – read VID_LOCK in the used video pipe(s) on deserializer
    g. Read register 0x1F to verify VPRBS_ERR_FLAG = 0 on deserializer
    h. Check VPRBS_FAIL = 0 and VPRBS_ERR = 0 in the used video pipe(s) on deserializer

To disable video PRBS testing, first disable the video PRBS checker in the deserializer, then disable the video PRBS generator in the serializer. Not forget to configure GMSL2 parts for normal operation again after disabling video PRBS testing if step 2-a is applied. Note that on the fly configuration is not supported. Thus, make sure video is stopped before configuring serializer and/or deserializer.

ADI Confidential for review under NDA
Analog Devices Confidential Distribute only under NDA
Provided by GMSL2 Customers Under NDA

43.7 Video Debug Procedure

1. Verify that the PCLKDET bit(s) in serializer are high once MIPI data is being transmitted. Note that the PCLKDET bit should be high for each used video pipe (typically one video pipe per data type sent from image sensor is used).
    a. If PCLKDET is not high, verify the following:
        i. Valid MIPI data is being sent to serializer.
        ii. Serializer is programmed to expect the correct number of lanes from image sensor (register 0x0331).
        iii. Serializer is programmed with the correct lane swapping from image sensor (registers 0x0332 and 0x0333.)
        iv. Serializer is programmed with the correct lane polarity from image sensor (register 0x0334).
        v. Serializer is correctly routing MIPI data from the MIPI controllers to the video pipes. Verify that registers 0x0314, 0x0315, 0x0316, 0x0317, 0x0318, 0x0319, 0x031A, and 0x031B are programmed to the correct value.
        vi. VS and HS routing is properly routed. Verify registers 0x3C0-0x3CF.

2. If the PCLKDET bit(s) is (are) high in the serializer, verify that the VID_LOCK bit(s) is (are) high in the deserializer. PCLKDET means that the serializer is most likely receiving and transmitting valid MIPI data to the deserializer. VID_LOCK means that the deserializer is most likely receiving valid MIPI data from the serializer and transmitting the MIPI data from the MIPI ports.
    a. If PCLKDET is high but VID_LOCK is low, verify the following:
        i. GMSL PHY is error free.
        ii. The video pipes in serializer are set to transmit data, and the video pipes in deserializer are set to receive data. Register 0x0002 in both the deserializer and serializer need to have each video pipe used enabled and each video pipe not used disabled.

3. If PCLKDET is high and VID_LOCK is high and the SoC is still not recognizing valid MIPI data, verify the following:
    a. MIPI transmitter is properly programmed to what the SOC is expecting.
        i. Number of lanes
        ii. Lane swapping
        iii. Lane polarity
        iv. MIPI clock speed
    b. Data routing internal to the deserializer is correct.
        i. Video pipe to MIPI controller programming. Reference the Deserializer Configuration section for additional information on how to properly program the deserializer.

4. Check deserializer CSI controller PLL lock.
    a. CSIPLLX/Y/Z/U_LOCKED – (register BACKTOP1) – this bit indicates if CSI output PLL is locked.

ADI Confidential for Review Only Under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

43.8 Feature Availability by Device Family

CSI-2 is available on several GMSL2 device families (*Table 230*).

Table 230. CSI-2 Support by Device Family

Product Family	Part Number
CSI-2 Serializers	MAX9295A MAX9295B
Advanced CSI-2 Serializers	MAX9295D MAX9295E
CSI-2 Camera Deserializers	MAX96784 MAX9296A MAX9296B MAX9296C
CSI-2 Display Deserializers	MAX96756 MAX96756F MAX96758F MAX96758
CSI-2 Quad Deserializers	MAX96712 MAX96712B MAX96722 MAX96722B

Note: For information regarding the CSI-2 Quad Deserializers, reference the CSI-2 Quad Deserializers section.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44 CSI-2 Quad Deserializers

44.1 Overview

GMSL2 CSI-2 quad deserializers receive data from up to four separate GMSL1 or GMSL2 serializers over 50Ω coax or 100Ω shielded twisted pair (STP) cables. The combined image data is converted to MIPI CSI-2 (D-PHY v1.2 up to 2.5Gbps/lane or C-PHY v1.0 up to 5.7Gbps/lane) and can be aggregated to any of the output MIPI ports.

There are two GMSL2 CSI-2 quad deserializer device variants: the MAX96712(B) and the MAX96722(B) (see note below). The MAX96712 operates each GMSL2 link at a fixed rate of 3Gbps or 6Gbps in the forward direction and 187.5Mbps in the reverse direction; the MAX96722 can operate GMSL2 links at 3Gbps in the forward direction and 187.5Mbps in the reverse direction. Both devices are GMSL1 backwards compatible and can pair with first-generation GMSL1 serializers running at rates up to 3.12Gbps (GMSL1 serial link data rate is variable and application-dependent). See Table 231 below.

Table 231. GMSL2 CSI-2 Quad Deserializer Link Rates

Part Number	GMSL2 Link Rates		GMSL1 Backward Compatible Mode	Reference over Reverse (RoR)
	Forward Channel	Reverse Channel		
MAX96712	6Gbps or 3Gbps	187Mbps	Up to 3.12Gbps*	No
MAX96712B	6Gbps or 3Gbps	187Mbps	Up to 3.12Gbps*	Yes
MAX96722	3Gbps	187Mbps	Up to 3.12Gbps*	No
MAX96722B	3Gbps	187Mbps	Up to 3.12Gbps*	Yes

* The GMSL1 serial link data rate is variable and application-dependent. The maximum data rate is 3.12Gbps.

Note: This section refers to the MAX96712/MAX96712B as the 'MAX96712' and the MAX96722/MAX96722B as the 'MAX96722'. These parts are functionally equivalent except for the inclusion of Reference over Reverse (RoR) in the B-type device variants in GMSL2 mode.

44.2 Operation

44.3 GMSL2 CSI-2 quad deserializers are a distinct device family from other GMSL2 CSI-2 deserializers with specific architecture and programming (detailed in this section). Reference the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

44.3.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:

Analog Devices Confidential, Distribute Outside of Maxim Integrated Company Without Prior Written Consent, Provided to GMSL2 Customers Under NDA
ADI Confidential for review under NDA

Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock, lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed

ADI Confidential for New vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

44.3.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for New vision under NDA
Analog Devices Confidential Document under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers under NDA

44.4 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 section for a description of the CSI-2 protocol.

44.4.1 Architecture

44.5 The following block diagrams show the video path within the GMSL2 CSI-2 quad deserializers' D-PHY (Figure 208 and Figure 209) and C-PHY (Figure 210 and Figure 211) architectures. Reference the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2.
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

44.5.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

ADI Confidential for New Vision under NDA
Maxim Integrated Confidential Property of Maxim Integrated Company Confidential Provided to CMSL2 Customers Under NDA

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock,

ADI Confidential Distribution Only Under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customer Under NDA

					lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

44.5.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

		watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for Newvision Under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.6 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 section for an explanation of the controllers and ports.

Note: The default lane mapping of GMSL2 CSI-2 quad deserializers does not match the pinout of the devices. This must be corrected by writing register 0x08A3 to 0x08E4 (see Table 238). After writing this register, the device will match the datasheet pin mapping. Reference MIPI PHY Settings for more information.

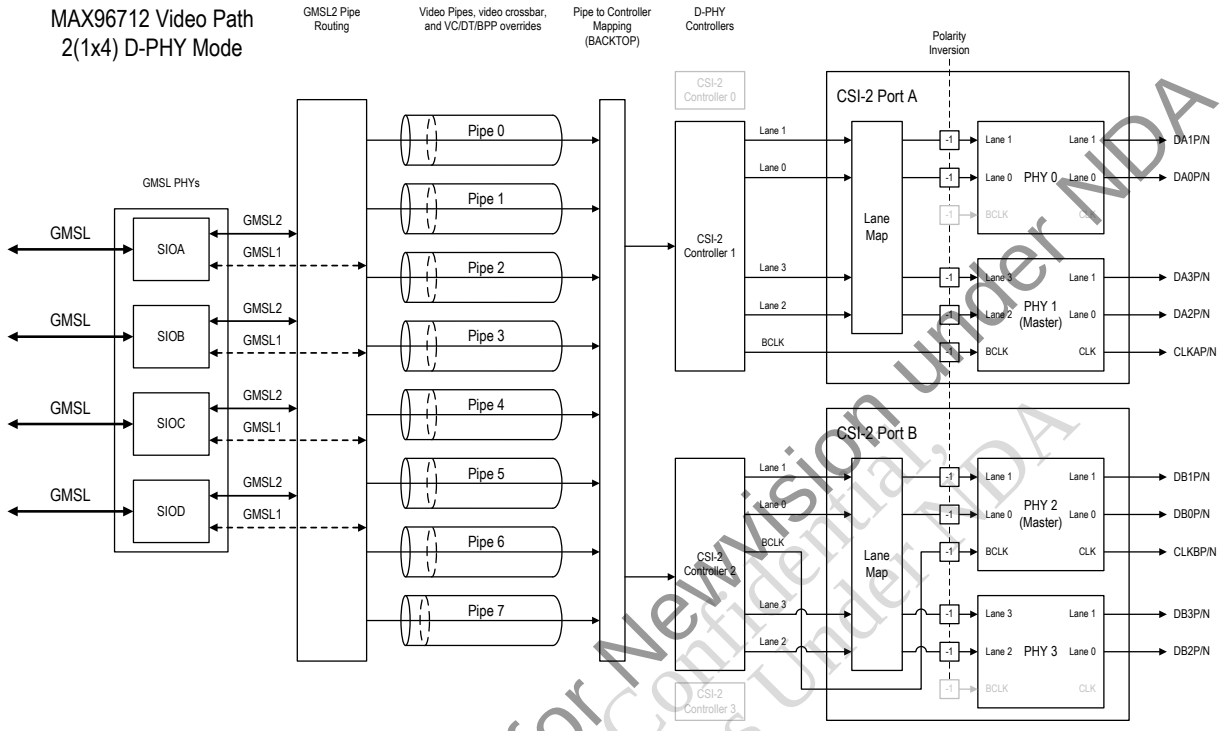


Figure 208. MAX96712 Video Path – 2(1x4) D-PHY Mode

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

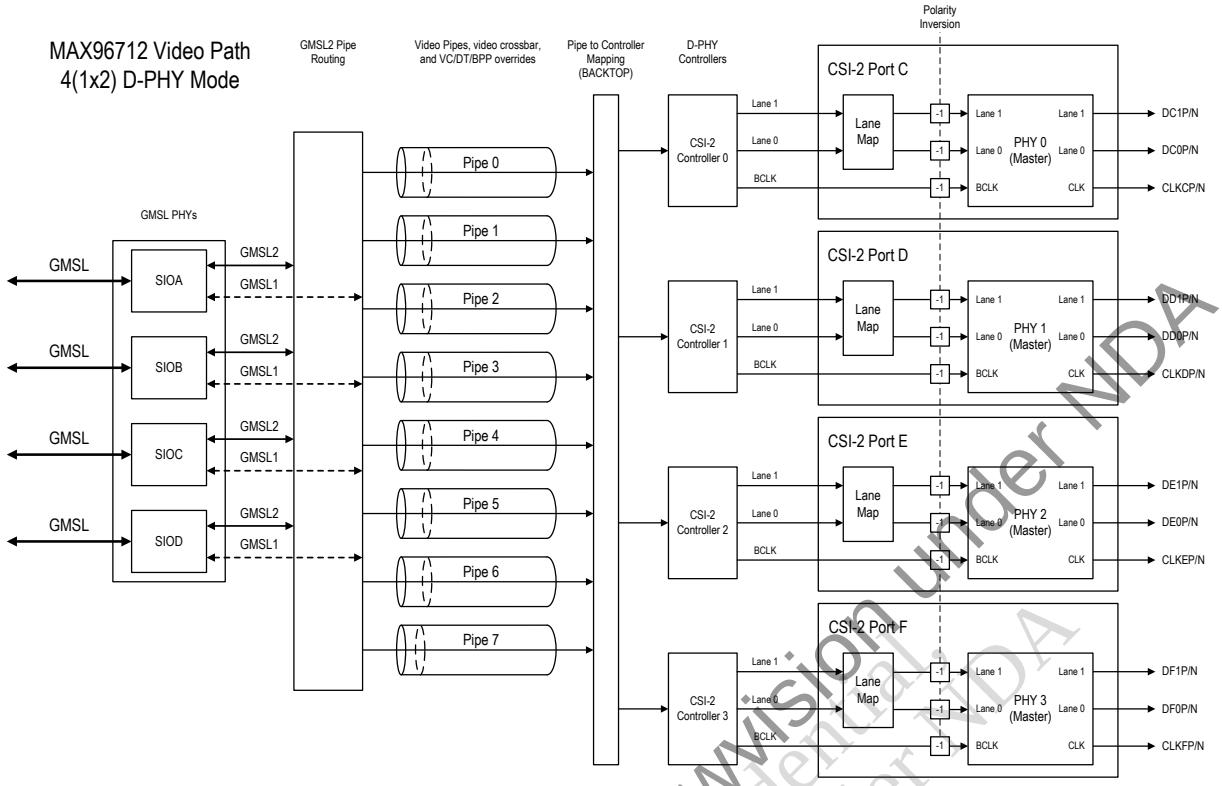


Figure 209. MAX96712 Video Path – 4(1x2) D-PHY Mode

ADI Confidential for New Revision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

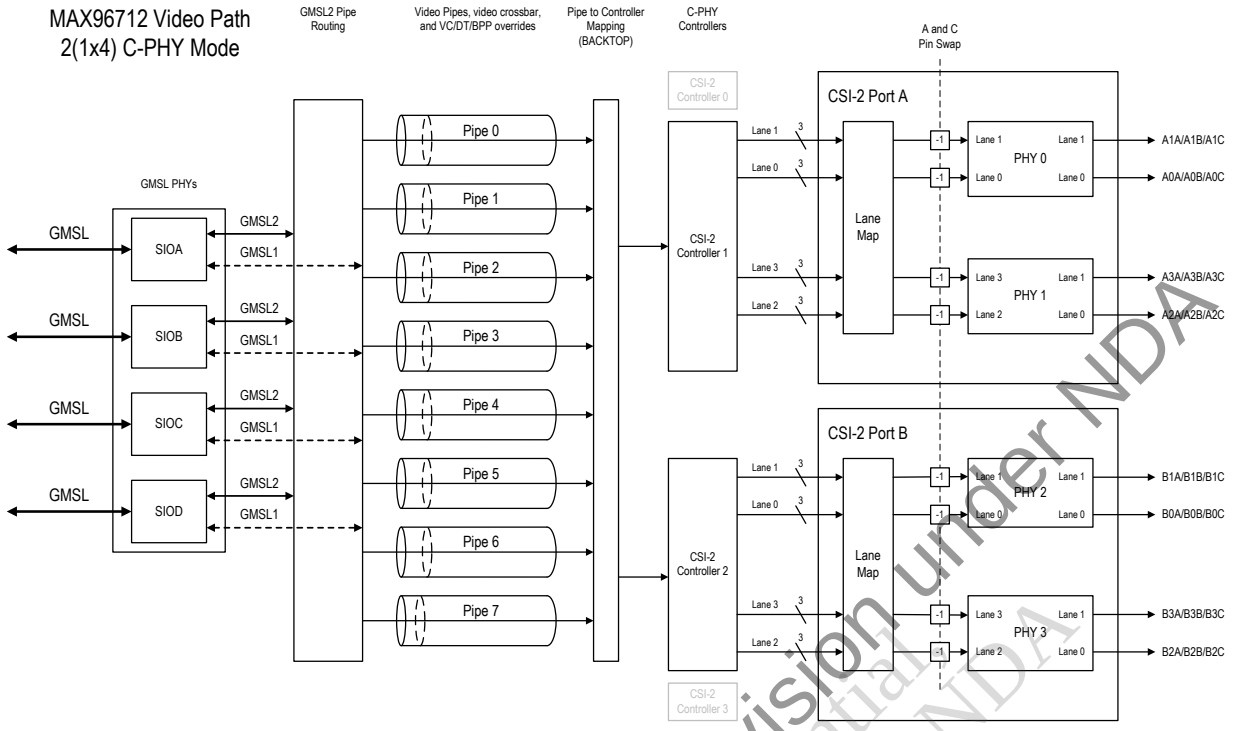


Figure 210.MAX96712 Video Path – 2(1x4) C-PHY Mode

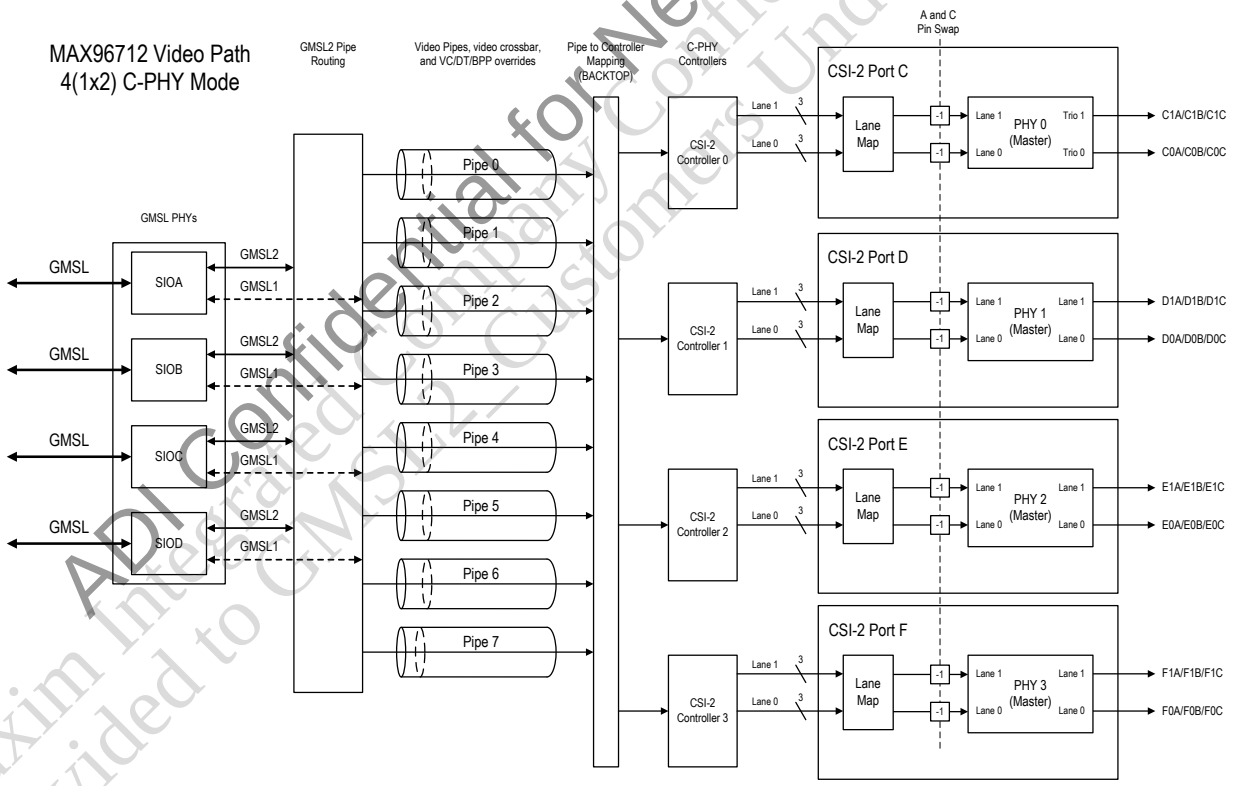


Figure 211. MAX96712 Video Path – 4(1x2) C-PHY Mode

ADI Confidential for New vision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.7 Configuration

The GMSL2 CSI-2 quad deserializer must be configured before receiving input from the serial link. Dynamic configuration is not supported.

The forward video path of GMSL2 CSI-2 quad deserializers is configured with the following programming:

- *Link Initialization*
- *Video Pipe Selection*
- *Video Pipe to MIPI Controller Mapping*
- *MIPI PHY Settings*

Only after the video path is configured should video be enabled. The following sub-sections detail the operation of each of these steps with descriptions of relevant registers and programming examples.

Note: In the 'Programming Examples' below, the slave device uses the address "0x52".

44.7.1 Link Initialization

Link initialization establishes the device link mode(s) and link speed(s). The GMSL2 CSI-2 quad deserializers have four independent GMSL links/PHYs (A, B, C, and D). The devices accept any combination of GMSL1 and GMSL2 inputs and link rates.

GMSL2 CSI-2 quad deserializers are powered up in either GMSL1 or GMSL2 mode for all links (through CFG1 pin) by default. After initial power-up, link mode(s) can be individually reconfigured via the GMSL2_A, GMSL2_B, GMSL2_C, and GMSL2_D bits (register 0x0006) followed by a one-shot link reset (register 0x0018). A short delay is required after each link reset before communication to the serializers is available (the link lock time is 45ms typical). Table 232 contains key link initialization registers.

Note: A link reset (register 0x0018) must be performed after mode/rate is changed for that setting to take effect. See Resets section for additional details.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

Table 232. Link Initialization Registers

Register	Bits	Default Value	Description
0x0006	7:0	0xFF	**GMSL Link/PHY Enable and Mode Select Register:** Bit 7: PHY D, 0 = GMSL1, 1 = GMSL2. Bit 6: PHY C, 0 = GMSL1, 1 = GMSL2 Bit 5: PHY B, 0 = GMSL1, 1 = GMSL2 Bit 4: PHY A, 0 = GMSL1, 1 = GMSL2 Bit 3: PHY D, 0 = Link Disable, 1 = Link Enable Bit 2: PHY C, 0 = Link Disable, 1 = Link Enable Bit 1: PHY B, 0 = Link Disable, 1 = Link Enable Bit 0: PHY A, 0 = Link Disable, 1 = Link Enable
0x0010	7:0	0x22	**GMSL Link/PHY Rate Select Register:** Bits [7:6]: Tx Rate on Link B Bits [5:4]: Rx Rate on Link B Bits [3:2]: Tx Rate on Link A Bits [1:0]: Rx Rate on Link A Tx Rate (Transmitter rate; i.e., reverse channel): • 00 = 187.5Mbps Rx Rate (Receiver rate; i.e., forward channel): • 01 = 3Gbps • 10 = 6Gbps
0x0011	7:0	0x22	**GMSL Link/PHY Rate Select Register:** Bits [7:6]: Tx Rate on Link D Bits [5:4]: Rx Rate on Link D Bits [3:2]: Tx Rate on Link C Bits [1:0]: Rx Rate on Link C Tx Rate (Transmitter rate; i.e., reverse channel): • 00 = 187.5Mbps Rx Rate (Receiver rate; i.e., forward channel): • 01 = 3Gbps • 10 = 6Gbps
0x0018	7:0	0x00	**GMSL Link Reset Register:** * Bits [7:4]: Link reset register for each link D/C/B/A 0: Release link reset 1: Activate link reset Bits [3:0]: One-shot link reset for each link D/C/B/A 0: No action 1: Reset data path (self-clear)

Note: A link reset on CSI-2 quad deserializers resets the entire data path of any video connected to the reset PHY. This behavior is different than other GMSL2 deserializers. Link resets should not be used when data is running in the device. Doing this may corrupt data and have unintended consequences.

ADI Confidential for NEW Version under NDA
Analog Devices Confidential, Provided to GMSL2_ Customer Under NDA
Maxim Integrated Company Confidential,

44.7.1.1 Link Lock Check

Pin #2 (MFP1) is used as LOCK indication by default. It is open-drain and will assert only when all enabled GMSL2 links (set by LINK_EN_A, LINK_EN_B, LINK_EN_C, and LINK_EN_D bits in register 0x0006) are locked. Unused links should be disabled on order to obtain the lock indicator and to facilitate data output. The following are lock indication register bits for each individual link:

- **GMSL1 Mode**
  - Register 0x0BCB, 0x0CCB, 0x0DCB or 0x0ECB bit 0 will assert if link A, B, C, or D is locked in GMSL1 mode.
- **GMSL2 Mode**
  - Register 0x001A, 0x000A, 0x000B, or 0x000C bit 3 will assert if link A, B, C, or D is locked in GMSL2 mode.

When pairing with GMSL2 serializers in GMSL2 mode, links are automatically locked upon connection. The GMSL2 protocol uses a fixed link rate based on a constant-frequency link clock generated from a 25MHz crystal oscillator or an external reference frequency. The link clock does not have any relationship with the input data pixel clock. In GMSL1 mode, the link is not automatically established and locked. If there is no pixel clock present on the serializer side, the reverse control link must be established first. Reference the *Pairing with GMSL1 Serializers* section for details.

44.7.1.1.1 Programming Example

This example enables link A and link B in GMSL2 mode.

```
# Turn on link A and B only under GMSL2 mode.
0x52,0x0006,0xF3
# Set 6Gbps/187.5Mbps for link A and B
0x52,0x0010,0x22
# Reset Oneshot for links A and B
0x52,0x0018,0x03
```

Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
ADI Confidential for Newvision under NDA

44.7.2 Video Pipe Selection

Video pipes must be configured to match the video streams received from connected serializers. This programming step, typically performed following link initialization, ensures that the GMSL2 CSI-2 quad deserializer properly receives video data from serializers. There are eight video pipes in total (video pipes 0–7).

Note: GMSL2 CSI-2 quad deserializers can receive video streams from a serializer in splitter mode; however, video streams must be routed to separate PHYs.

Figure 212 shows the relationship between GMSL1 and GMSL2 data and the video pipes. Video pipe selection registers are presented in Table 233.

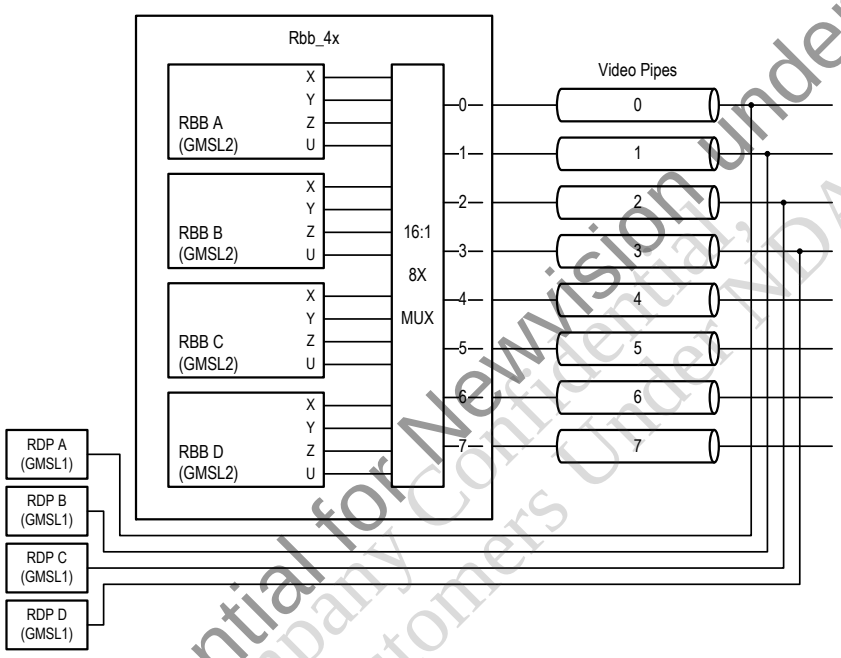


Figure 212. Video Pipe Selection Block Diagram

Note: Reference the Video Data Replication section below and Data Replication section in the main CSI-2 section for details on configuring video pipe data replication.

44.7.2.1 GMSL1 Mode

By default, video data received from PHY A, B, C, and D is output after video pipes 0, 1, 2, and 3, respectively. In GMSL1 mode, video pipe selection is not required; however, at least one video pipe must be enabled.

Note: Video pipes 0, 1, 2, and 3 are enabled by default.

44.7.2.2 GMSL2 Mode

The 16:1 MUX must be configured to match the video streams (STR_ID) from serializers at each video pipe. Note that most GMSL2 camera serializers have four video pipes (X, Y, Z, and U).

Analog Devices Confidential for New Revision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

Table 233. Video Pipe Selection Registers

Register	Bits	Default Value	Description
0x00F0	7:0	0x62	**Video Pipe Select Register:** Bits [7:6]: PHY Selection for Pipe 1 Bits [5:4]: Stream ID Selection for Pipe 1 Bits [3:2]: PHY Selection for Pipe 0 Bits [1:0]: Stream ID Selection for Pipe 0 PHY (Link) Selection: • 00 = PHY A • 01 = PHY B • 10 = PHY C • 11 = PHY D Serializer Stream ID Selection: • 00 = Stream 0 • 01 = Stream 1 • 10 = Stream 2 • 11 = Stream 3
0x00F1	7:0	0xEA	**Video Pipe Select Register:** Bits [7:6]: PHY Selection for Pipe 3 Bits [5:4]: Stream ID Selection for Pipe 3 Bits [3:2]: PHY Selection for Pipe 2 Bits [1:0]: Stream ID Selection for Pipe 2 PHY (Link) Selection: • 00 = PHY A • 01 = PHY B • 10 = PHY C • 11 = PHY D Serializer Stream ID Selection: • 00 = Stream 0 • 01 = Stream 1 • 10 = Stream 2 • 11 = Stream 3
0x00F2	7:0	0x40	**Video Pipe Select Register:** Bits [7:6]: PHY Selection for Pipe 5 Bits [5:4]: Stream ID Selection for Pipe 5 Bits [3:2]: PHY Selection for Pipe 4 Bits [1:0]: Stream ID Selection for Pipe 4 PHY (Link) Selection: • 00 = PHY A • 01 = PHY B • 10 = PHY C • 11 = PHY D Serializer Stream ID Selection: • 00 = Stream 0 • 01 = Stream 1 • 10 = Stream 2 • 11 = Stream 3
0x00F3	7:0	0xC8	**Video Pipe Select Register:** Bits [7:6]: PHY Selection for Pipe 7 Bits [5:4]: Stream ID Selection for Pipe 7 Bits [3:2]: PHY Selection for Pipe 6

ADI Confidential for New Version under NDA
Maxim Integrated Confidential Document
Provided to GMSL2_Customer's Company Employee Under NDA

			Bits [1:0]: Stream ID Selection for Pipe 6 PHY (Link) Selection: • 00 = PHY A • 01 = PHY B • 10 = PHY C • 11 = PHY D Serializer Stream ID Selection: • 00 = Stream 0 • 01 = Stream 1 • 10 = Stream 2 • 11 = Stream 3
0x00F4	7:0	0x0F	**Video Pipe Enable Register:** Bit 7: Enable video pipe 7 Bit 6: Enable video pipe 6 Bit 5: Enable video pipe 5 Bit 4: Enable video pipe 4 Bit 3: Enable video pipe 3 Bit 2: Enable video pipe 2 Bit 1: Enable video pipe 1 Bit 0: Enable video pipe 0 0 = Disable; 1 = Enable

44.7.2.3 Video Lock Check

Register 0x01DC, 0x01FC, 0x021C, 0x023C, 0x025C, 0x027C, 0x029C, or 0x02BC bit 0 (VIDEO_LOCK) for each of the eight video pipes will assert if it is receiving valid video data.

Note: VIDEO_LOCK in GMSL1 mode is equivalent to serial Link Lock (see Link Lock Check).

44.7.2.3.1 Programming Example

This example sets up the deserializer video pipes 0-3 to have link A and B video streams.

```
# Enable pipe 0 for link A stream ID 0, pipe 1 for link B stream ID 0
0x52,0x00F0,0x40
# Enable pipe 2 for link A stream ID 1, pipe 3 for link B stream ID 1
0x52,0x00F1,0x51
# Turn on pipe 0, 1, 2 and 3
0x52,0x00F4,0x0F
```

ADI Confidential for Newvision under NDA
Analog Confidential, Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2 Customers Under NDA

44.7.3 Video Pipe to MIPI Controller Mapping

Video pipe to MIPI controller mapping is used to reconfigure the default data paths and route streams to different MIPI ports. GMSL2 CSI-2 quad deserializers have four MIPI controllers (Ctrl 0, 1, 2, and 3) that control four different MIPI PHYs (MIPI PHY 0, 1, 2, and 3).

The necessity of this step is dependent upon the serial link connection(s). If the device has only one serial link connection, the step may be skipped; however, this step is often necessary if more than one serial link is established with the quad deserializer. In this case, video data in multiple video pipes can be aggregated or concatenated into one or more MIPI port(s). Table 234 contains video pipe to MIPI PHY controller registers.

The default video pipe MIPI controller connections are shown below for GMSL1 and GMSL2 modes.

44.7.3.1 GMSL1 Mode

- GMSL1 PHY A → (Video Pipe 0) → Controller 0 / MIPI PHY 0
- GMSL1 PHY B → (Video Pipe 1) → Controller 1 / MIPI PHY 1
- GMSL1 PHY C → (Video Pipe 2) → Controller 2 / MIPI PHY 2
- GMSL1 PHY D → (Video Pipe 3) → Controller 3 / MIPI PHY 3

Note: There is no “video pipe” concept in GMSL1 links. The data path can be considered a direct link with the MIPI controller although controller mapping is supported in GMSL1 mode as well as aggregation.

44.7.3.2 GMSL2 Mode

- Video Pipe 0 → Controller 0 / MIPI PHY 0
- Video Pipe 1 → Controller 1 / MIPI PHY 1
- Video Pipe 2 → Controller 2 / MIPI PHY 2
- Video Pipe 3 → Controller 3 / MIPI PHY 3
- Video Pipe 4 → Controller 0 / MIPI PHY 0
- Video Pipe 5 → Controller 1 / MIPI PHY 1
- Video Pipe 6 → Controller 2 / MIPI PHY 2
- Video Pipe 7 → Controller 3 / MIPI PHY 3

The following steps describe the process of configuring video pipe to MIPI controller mapping (*Figure 213*). First, enable the number of mappings required (of a maximum possible 16) by writing the two map enable registers (MAP_EN_L and MAP_EN_H) for each video pipe to be configured. Then, specify the destination for each enabled mapping block (i.e., the MIPI PHY controller destination). The data being mapped is designated via programming two register sets for the source virtual channel (VC) and data type (DT) as well as the destination VC and DT. By setting the destination values different than the source, the VC and DT can be changed.

Note: When pairing with serializers in parallel mode, software overrides in the quad deserializer may be required prior to mapping. This is discussed in the *Software Override* section below.

Maxim Integrated Confidential, Provided to GMSL2_ Customers Under NDA
Analog Devices Confidential and proprietary. Not to be distributed outside ADI without a signed NDA
ADI Confidential for New Vision under NDA

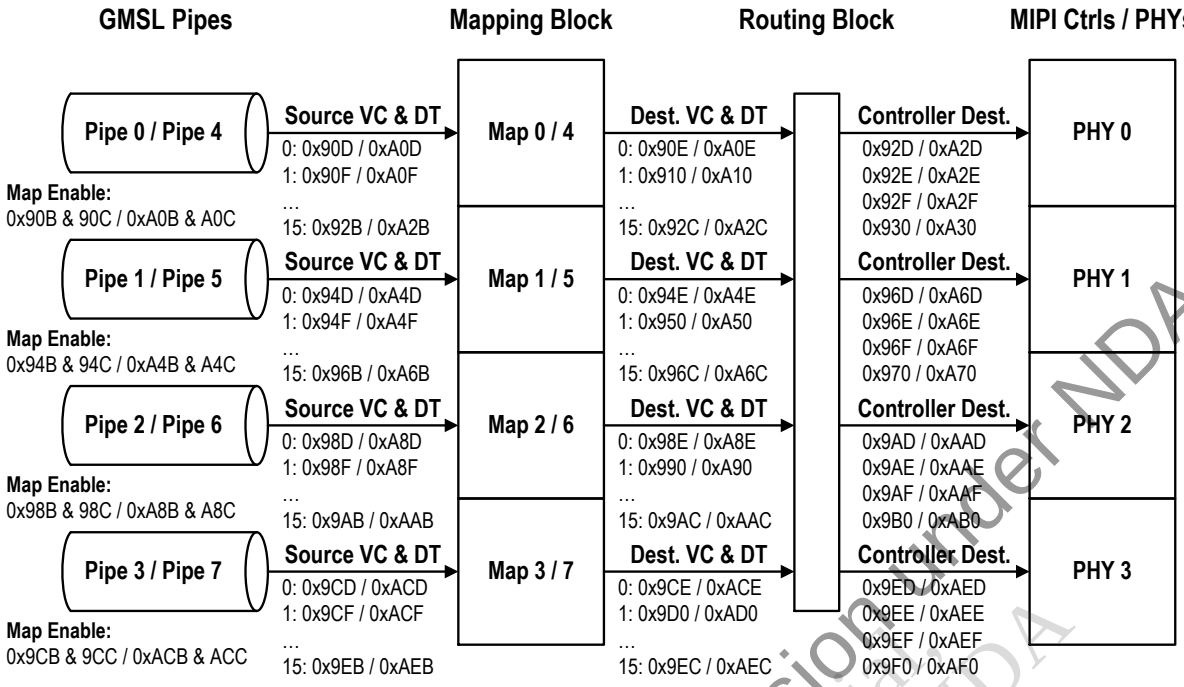


Figure 213. Video Pipe to MIPI Controller Mapping Block Diagram

Under each MIPI TX register block:

- Video Pipe 0 mapping register block: 0x090B – 0x0930
- Video Pipe 1 mapping register block: 0x094B – 0x0970
- Video Pipe 2 mapping register block: 0x098B – 0x09B0
- Video Pipe 3 mapping register block: 0x09CB – 0x09F0
- Video Pipe 4 mapping register block: 0x0A0B – 0x0A30
- Video Pipe 5 mapping register block: 0x0A40 – 0x0A70
- Video Pipe 6 mapping register block: 0x0A80 – 0x0AB0
- Video Pipe 7 mapping register block: 0x0AC8 – 0x0AF0

Table 234. Video Pipe to MIPI PHY Controller Registers

Register	Bits	Default Value	Description
0x090B	7:0	0x00	**Mapping Enable Low Byte Register:** 0000_0000 = No mapping enabled. XXXX_XXX1 = Map SRC_0 to DES_0. … 1XXX_XXXX = Map SRC_7 to DES_7 Each bit enables 1 of max 16 mapping and distribution entries for current video stream.
0x090C	7:0	0x00	**Mapping Enable High Byte Register:** 0000_0000 = No mapping enabled. XXXX_XXX1 = Map SRC_8 to DES_8. … 1XXX_XXXX = Map SRC_15 to DES_15 Each bit enables 1 of max 16 mapping and distribution entries for current video stream.

ADI Confidential Information Under NDA
Maxim Integrated Official Distribution
Provided to GMSL2 Customers Under NDA
Company Confidential

0x090D	7:0	0x00	**MAP_SRC_0 Register:** Bits [7:6]: VC Bits [5:0]: DT
0x090E	7:0	0x00	**MAP_DES_0 Register:** Bits [7:6]: VC Bits [5:0]: DT
…	…	…	…
0x092B	7:0	0x00	**MAP_SRC_15 Register:** Bits [7:6]: VC Bits [5:0]: DT
0x092C	7:0	0x00	**MAP_DES_15 Register:** Bits [7:6]: VC Bits [5:0]: DT
0x092D	7:0	0x00	**MAP Destination Controller Register:** Bits [7:6]: Destination Controller for Map 3 Bits [5:4]: Destination Controller for Map 2 Bits [3:2]: Destination Controller for Map 1 Bits [1:0]: Destination Controller for Map 0 00 = Map to Controller 0 01 = Map to Controller 1 10 = Map to Controller 2 11 = Map to Controller 3
0x092E	7:0	0x00	**MAP Destination Controller Register:** Bits [7:6]: Destination Controller for Map 7 Bits [5:4]: Destination Controller for Map 6 Bits [3:2]: Destination Controller for Map 5 Bits [1:0]: Destination Controller for Map 4 00 = Map to Controller 0 01 = Map to Controller 1 10 = Map to Controller 2 11 = Map to Controller 3
0x092F	7:0	0x00	**MAP Destination Controller Register:** Bits [7:6]: Destination Controller for Map 11 Bits [5:4]: Destination Controller for Map 10 Bits [3:2]: Destination Controller for Map 9 Bits [1:0]: Destination Controller for Map 8 00 = Map to Controller 0 01 = Map to Controller 1 10 = Map to Controller 2 11 = Map to Controller 3
0x0930	7:0	0x00	**MAP Destination Controller Register:** Bits [7:6]: Destination Controller for Map 15 Bits [5:4]: Destination Controller for Map 14 Bits [3:2]: Destination Controller for Map 13 Bits [1:0]: Destination Controller for Map 12 00 = Map to Controller 0 01 = Map to Controller 1 10 = Map to Controller 2 11 = Map to Controller 3

ADI Confidential for Review in under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customer under NDA
Device Confidential, Provided under NDA

44.7.3.2.1 Programming Examples

This example enables video pipe mapping for data, sets the destination controller, and individually programs the source and destination data type and virtual channel for each.

```
# Enable 3 mappings for data in video pipe 0
0x52,0x090B,0x07
# Set all 3 mapping destinations to be MIPI PHY Controller 1
0x52,0x092D,0x15
# Set source and destination VC/DT for 3 data types (RAW12, FS and FE) individually, and
change VC from 0 to 1
0x52,0x090D,0x2C
0x52,0x090E,0x6C
0x52,0x090F,0x00
0x52,0x0910,0x40
0x52,0x0911,0x01
0x52,0x0912,0x41
```

The deserializer receives data on links A and B from serializer video pipe X in this example. In the deserializer, the RAW12 data in video pipes 1 and 2 is aggregated to PHY1 output.

```
# Selects Link A and Link B, pipe X, for pipe 0 and pipe 1.
0x52, 0xF0, 0x40
# Only turn on pipes 0-1
0x52, 0xF4, 0x03

# Enable mapping for pipe 0, 3 mappings enabled for FS, FE, and data type
0x52, 0x90B, 0x07
# Map_SRC_0 - FS, VC = 0
0x52, 0x90D, 0x00
# Map_DST_0 - FS, VC = 0
0x52, 0x90E, 0x00
# Map_SRC_1 - Raw12, VC = 0
0x52, 0x90F, 0x2C
# Map_DST_1 - Raw12, VC = 0
0x52, 0x910, 0x2C
# Map_SRC_2 - FE, VC = 0
0x52, 0x911, 0x01
# Map_DST_2 - FE, VC = 0
0x52, 0x912, 0x01
# Map_D-PHY_DST Sets PHY destination to PHY1 for mappings 0-2
0x52, 0x92D, 0x15

# Enable mapping for pipe1, 3 mappings enabled for FS, FE, and data type
0x52, 0x94B, 0x07
# Map_SRC_0 - FS, VC = 0
0x52, 0x94D, 0x00
# Map_DST_0 - FS, VC = 1
0x52, 0x94E, 0x40
# Map_SRC_1 - Raw12, VC = 0
0x52, 0x94F, 0x2C
# Map_DST_1 - Raw12, VC = 1
0x52, 0x950, 0x6C
# Map_SRC_2 - FE, VC = 0
0x52, 0x951, 0x01
# Map_DST_2 - FE, VC = 1
0x52, 0x952, 0x41
# Map_D-PHY_DST Sets PHY destination to PHY1 for mappings 0-2
0x52, 0x96D, 0x15
```

ADI Confidential for New vision under NDA
Maxim Confidential, Distribute to Quanta under NDA
Property of Maxim Integrated Company Confidential, Distribute to GMSL2 Customers Under NDA

44.7.3.3 Extended Virtual Channels

GMSL2 quad deserializers can use CSI-2 extended virtual channels to accommodate larger serial link systems. Virtual channels allow the serial link system to differentiate video inputs by the virtual channel assigned to them. When extended virtual channels is enabled, the standard 2-bit virtual channel selection is extended to 4-bit (D-PHY) or 5-bit (C-PHY), increasing the number of available virtual channels to 16 for D-PHY applications and 32 for C-PHY applications. The increase in available virtual channels allows systems to support more camera inputs.

Extended virtual channels are enabled individually for each of the four controllers (0-3) on the quad deserializer. Any video pipes routed to a controller with virtual channel extension enabled will use the extended virtual channels.

Reference Table 235 for extended virtual channels registers details.

Note: The two virtual channel source-to-destination mapping bits referenced in Table 234 are used with extended virtual channel configuration; however, these bits become the least significant bits (LSB) of the virtual channel bit selection.

Table 235. Extended Virtual Channels Registers

Register	Bits	Default Value	Description
0x090A/0x094A/0x098A/0x09CA	4	0	**CSI2_vcx_en[?]:** 0 = VC extension Disabled 1 = VC extension Disabled Controller 0,1,2, and 3 virtual channel enable.
0x800	7:2	0x00	**Pipe 0, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x80F	7:2	0x00	**Pipe 0, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x810	7:2	0x00	**Pipe 1, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x81F	7:2	0x00	**Pipe 1, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x820	7:2	0x00	**Pipe 2, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x82F	7:2	0x00	**Pipe 2, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x830	7:2	0x00	**Pipe 3, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel

ADI Confidential for New vision under NDA
Analog Devices Confidential, Provided to Maxim Integrated Company Confidential, GMSL2_Customers Under NDA

			Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x83F	7:2	0x00	**Pipe 3, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x840	7:2	0x00	**Pipe 4, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x84F	7:2	0x00	**Pipe 4, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x850	7:2	0x00	**Pipe 5, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x85F	7:2	0x00	**Pipe 5, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x860	7:2	0x00	**Pipe 6, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x86F	7:2	0x00	**Pipe 6, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
0x870	7:2	0x00	**Pipe 7, Extended VC MAP_SRC/DST_0 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel
…	…	…	**…**
0x87F	7:2	0x00	**Pipe 7, Extended VC MAP_SRC/DST_15 Register:** Bits [7:5]: SRC Virtual Channel Bits [4:2]: DST Virtual Channel

ADI Confidential for Newvision Under NDA
Analog Devices Confidential Document
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Under_NDA

44.7.3.3.1 Register Example

This example demonstrates the MSB and LSB of the extended virtual channel for pipe 0, mapping 0. Here, the source is changed from VC = 0 to destination VC = 5, VC = 4, or VC = 2.

Table 236. Source Mapping (Extended VC)

	EXT VC Source			VC Source	
Register	**0x0800 [7:5]**			**0x090D [7:6]**	
MIPI	**C-PHY Only**	**D-PHY and C-PHY**			
Bit Place	**4**	**3**	**2**	**1**	**0**
VC = 0	0	0	0	0	0

Table 237. Destination Mapping (Extended VC)

	EXT VC Destination			VC Destination	
Register	**0x0800 [4:2]**			**0x090E [7:6]**	
MIPI	**C-PHY Only**	**D-PHY and C-PHY**			
Bit Place	**4**	**3**	**2**	**1**	**0**
VC = 5	0	0	1	0	1
VC = 4	0	0	1	0	0
VC = 2	0	0	0	1	0

ADI Confidential for New vision under NDA
Analog Devices Confidential, Distribution Limited under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.7.3.3.2 Programming Example

This example routes video pipe 4 to controller 1 and assigns the extended virtual channel (VC = 4).

```
# Setup Source to Destination mappings
# Turn on controller 1 csi2_vcx_en
0x52, 0x94A, 0xD0
# Enable pipe 4, mapping 0-3.
0x52, 0xA0B, 0x07

# Mapping 0
# Source, VC = 0 and DT = FS
0x52, 0xA0D, 0x00
# Destination, VC = 0 and DT = FS, The 2 - VC bits here are the LSB part of the VC
bits for the VC extension.
0x52, 0xA0E, 0x00

# Mapping 1
# Source, VC = 0 and DT = Raw12
0x52, 0xA0F, 0x2C
# Destination, VC = 0 and DT = Raw12, The 2 - VC bits here are the LSB part of the
VC bits for the VC extension.
0x52, 0xA10, 0x2C

# Mapping 2
# Source, VC = 0 and DT = FE
0x52, 0xA11, 0x01
# Destination, VC = 0 and DT = FE, The 2 - VC bits here are the LSB part of the VC
bits for the VC extension.
0x52, 0xA12, 0x01

# D-PHY Destination is set to controller 1 for mapping 0-3
0x52, 0xA2D, 0x15

# Setup for Extended virtual channels
# Set the 3 MSB for the destination mapping. This is the MSB for mapping 0.
0x52, 0x840, 0x04
# Set the 3 MSB for the destination mapping. This is the MSB for mapping 1.
0x52, 0x841, 0x04
# Set the 3 MSB for the destination mapping. This is the MSB for mapping 2.
0x52, 0x842, 0x04
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.7.4 MIPI PHY Settings

The MIPI PHY settings contain programming options for output data rate, number of lanes, and port selection. The GMSL2 CSI-2 quad deserializers have four independent two-lane MIPI PHYs (PHY 0, 1, 2, and 3) controlled by four respective MIPI controllers (Ctrl 0, 1, 2, and 3). Table 238 contains the MIPI PHY settings registers. Reference the Architecture section above for block diagrams.

There are five MIPI PHY modes available for either D-PHY or C-PHY applications. Set one of the bits (exclusively) in register 0x08A0[4:0] to select the mode.

- 0x08A0 bit 0: 4 x 2 Mode
- 0x08A0 bit 1: 1 x 4 Mode (Ctrl 2 is master)
- 0x08A0 bit 2: 2 x 4 Mode (Ctrl 1 is the master for port A; Ctrl 2 is the master for port B)
- 0x08A0 bit 3: 1 x 4a + 2 x 2 Mode (Ctrl 1 is the master for port A)
- 0x08A0 bit 4: 1 x 4b + 2 x 2 Mode (Ctrl 2 is the master for port B)

In **x2 Mode**, four MIPI PHYs work independently. Each corresponding port (C, D, E, and F) has its own clock and two data lanes in D-PHY mode. Similarly, each port has two trios in C-PHY mode.

In **x4 Mode**, two of the MIPI PHYs are combined to establish 4-lane mode. In D-PHY mode, by default, PHY 1 and PHY 2 are the master PHYs providing the MIPI clock for ports A and B. The clock from slave PHYs can be used as the master clock by forcing the PHY 0 or 3 clock output on register 0x08A0. In C-PHY mode, the two-trio PHYs are combined into a four-trio.

The 1x1, 1x2, 1x3, and 1x4 configurations use lane count settings. For stability enhancement, it is recommended to always put the DPLL blocks in reset (config_soft_rst_n = 0) before changing the MIPI lane rate. Refer to the *Programming Example* below for register write details.

Table 238. MIPI PHY Settings Registers

Register	Bits	Default Value	Description
0x08A0	7:0	0x04	**MIPI PHY Mode Select Register:** Bit 7: Set to force all MIPI clocks running. Bit 6: Set to force PHY 3 MIPI clock running. Bit 5: Set to force PHY 0 MIPI clock running. Bit 4: MIPI PHY 1x4b + 2x2 Mode. Bit 3: MIPI PHY 1x4a + 2x2 Mode. Bit 2: MIPI PHY 2x4 Mode. Bit 1: MIPI PHY 1x4 Mode. Bit 0: MIPI PHY 4x2 Mode.
0x08A2	7:4	0xF0	**MIPI PHY Enable Register:** Bit 7: Enable MIPI PHY 3 Bit 6: Enable MIPI PHY 2 Bit 5: Enable MIPI PHY 1 Bit 4: Enable MIPI PHY 0 0 = PHY in standby 1 = PHY enabled.
0x08A3	7:0	0x4E	**MIPI PHY 0 and 1 Lane Mapping Register:** Bits [7:6]: Set PHY 1 Data Lane 1 Bits [5:4]: Set PHY 1 Data Lane 0 Bits [3:2]: Set PHY 0 Data Lane 1 Bits [1:0]: Set PHY 0 Data Lane 0

ADI Confidential for Review Under NDA
Analog Devices Confidential Distribution Only Under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

			00 = Map D0. 01 = Map D1. 10 = Map D2. 11 = Map D3.
0x08A4	7:0	0xE4	**MIPI PHY 3 and 2 Lane Mapping Register:** Bits [7:6]: Set PHY 3 Data Lane 1 Bits [5:4]: Set PHY 3 Data Lane 0 Bits [3:2]: Set PHY 2 Data Lane 1 Bits [1:0]: Set PHY 2 Data Lane 0 00 = Map D0. 01 = Map D1. 10 = Map D2. 11 = Map D3.
0x08A5	5:0	000000	**MIPI PHY 0 and 1 Lane Polarity Register:** Bit 5: Set Polarity on PHY 1 CLK lane Bit 4: Set Polarity on PHY 1 D1 lane / Swap PHY 1 D1 lane A & C pins in C-PHY mode Bit 3: Set Polarity on PHY 1 D0 lane / Swap PHY 1 D0 lane A & C pins in C-PHY mode Bit 2: Set Polarity on PHY 0 CLK lane Bit 1: Set Polarity on PHY 0 D1 lane / Swap PHY 0 D1 lane A & C pins in C-PHY mode Bit 0: Set Polarity on PHY 0 D0 lane / Swap PHY 0 D0 lane A & C pins in C-PHY mode 0 = normal polarity, 1 = inversed polarity.
0x08A6	5:0	000000	**MIPI PHY 3 and 2 Lane Polarity Register:** Bit 5: Set Polarity on PHY 3 CLK lane Bit 4: Set Polarity on PHY 3 D1 lane/ Swap PHY 3 D1 lane A & C pins in C-PHY mode Bit 3: Set Polarity on PHY 3 D0 lane/ Swap PHY 3 D0 lane A & C pins in C-PHY mode Bit 2: Set Polarity on PHY 2 CLK lane Bit 1: Set Polarity on PHY 2 D1 lane/ Swap PHY 2 D1 lane A & C pins in C-PHY mode Bit 0: Set Polarity on PHY 2 D0 lane/ Swap PHY 2 D0 lane A & C pins in C-PHY mode 0 = normal polarity, 1 = inversed polarity.
0x08C9	3:0	0x00	**MIPI Controller Reset Register:** Bits [3:0]: resets for each controller 0/1/2/3 0: Do not reset 1: Reset the MIPI Controller
0x090A	7:5	110	**MIPI PHY 0 Lane Count / C-PHY enable Register:** Bits [7:6]: Set lane count. 00 = 1 data lane 01 = 2 data lanes 10 = 3 data lanes 11 = 4 data lanes Bit 5: Enable C-PHY; D-PHY mode by default. 0 = C-PHY disabled, 1 = C-PHY enabled.
0x094A	7:6	11	**MIPI PHY 1 Lane Count / C-PHY enable Register:** Bits [7:6]: Set lane count. 00 = 1 data lane

ADI Confidential for review under NDA
Analog Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

			01 = 2 data lane 10 = 3 data lane 11 = 4 data lane Bit 5: Enable C-PHY; D-PHY mode by default. 0 = C-PHY disabled, 1 = C-PHY enabled.
0x098A	7:6	11	**MIPI PHY 2 Lane Count / C-PHY enable Register:** Bits [7:6]: Set lane count. 00 = 1 data lane 01 = 2 data lane 10 = 3 data lane 11 = 4 data lane Bit 5: Enable C-PHY; D-PHY mode by default. 0 = C-PHY disabled, 1 = C-PHY enabled.
0x09CA	7:6	11	**MIPI PHY 3 Lane Count / C-PHY enable Register:** Bits [7:6]: Set lane count. 00 = 1 data lane 01 = 2 data lane 10 = 3 data lane 11 = 4 data lane Bit 5: Enable C-PHY; D-PHY mode by default. 0 = C-PHY disabled, 1 = C-PHY enabled.
0x0415	4:0	01111	**MIPI PHY 0 DPLL Freq Register:** Set DPLL frequency on multiple of 100MHz. **D-PHY** Clock freq is half; Data rate is equivalent bps/lane. 00010 = 200MHz DPLL, 200Mbps/lane data rate. … 11001 = 2500MHz DPLL, 2.5Gbps/lane data rate. **C-PHY** 2.28bits/symbol. 00010 = 200MHz DPLL, 456Mbps/lane data rate. … 11001 = 2500MHz DPLL, 5.7Gbps/lane data rate.
0x0418	4:0	01111	**MIPI PHY 1 DPLL Freq Register:** Set DPLL frequency on multiple of 100MHz. **D-PHY** Clock freq is half; Data rate is equivalent bps/lane. 00001 = 100MHz DPLL, 100Mbps/lane data rate. … 11001 = 2500MHz DPLL, 2.5Gbps/lane data rate. **C-PHY** 2.28bits/symbol. 00010 = 200MHz DPLL, 456Mbps/lane data rate. … 11001 = 2500MHz DPLL, 5.7Gbps/lane data rate.
0x041B	4:0	01111	**MIPI PHY 2 DPLL Freq Register:** Set DPLL frequency on multiple of 100MHz. **D-PHY** Clock freq is half; Data rate is equivalent bps/lane. 00001 = 100MHz DPLL, 100Mbps/lane data rate. … 11001 = 2500MHz DPLL, 2.5Gbps/lane data rate.

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

			C-PHY 2.28bits/symbol. 00010 = 200MHz DPLL, 456Mbps/lane data rate. … 11001 = 2500MHz DPLL, 5.7Gbps/lane data rate.
0x041E	4:0	01111	**MIPI PHY 3 DPLL Freq Register:** Set DPLL frequency on multiple of 100MHz. **D-PHY** Clock freq is half; Data rate is equivalent bps/lane. 00001 = 100MHz DPLL, 100Mbps/lane data rate. … 11001 = 2500MHz DPLL, 2.5Gbps/lane data rate. **C-PHY** 2.28bits/symbol. 00010 = 200MHz DPLL, 456Mbps/lane data rate. … 11001 = 2500MHz DPLL, 5.7Gbps/lane data rate.
0x1C00	0	1	**Controller 0 DPLL Register:** Set DPLL block into reset mode for Controller 0 0 = DPLL block in reset 1 = DPLL block in normal mode (default)
0x1D00	0	1	**Controller 1 DPLL Register:** Set DPLL block into reset mode for Controller 1 0 = DPLL block in reset 1 = DPLL block in normal mode (default)
0x1E00	0	1	**Controller 2 DPLL Register:** Set DPLL block into reset mode for Controller 2 0 = DPLL block in reset 1 = DPLL block in normal mode (default)
0x1F00	0	1	**Controller 3 DPLL Register:** Set DPLL block into reset mode for Controller 3 0 = DPLL block in reset 1 = DPLL block in normal mode (default)

ADI Confidential for review under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_ Customer under NDA

44.7.4.1 Lane Swap

Lane swapping is available for pins on the same port in both C-PHY and D-PHY modes. Reference Table 238 for relevant registers.

Note: The default lane mapping of GMSL2 CSI-2 quad deserializers does not match the pinout of the devices. This must be corrected with lane swapping by writing register 0x8A3 to 0hE4 (see Table 238).

44.7.4.1.1 D-PHY Lane Swap Example

For D-PHY output, the data pins can be swapped within each port, but the clock location is fixed. For example, in 2x4 mode, the default mappings of the D0, D1, D2, and D3 pairs can be swapped to different output pins. Additionally, the polarity of each output data pair and the clock lane support polarity inversion. Figure 214 shows the lane swap required to match the device pinout in D-PHY mode. Figure 215 demonstrates polarity swap in D-PHY mode.

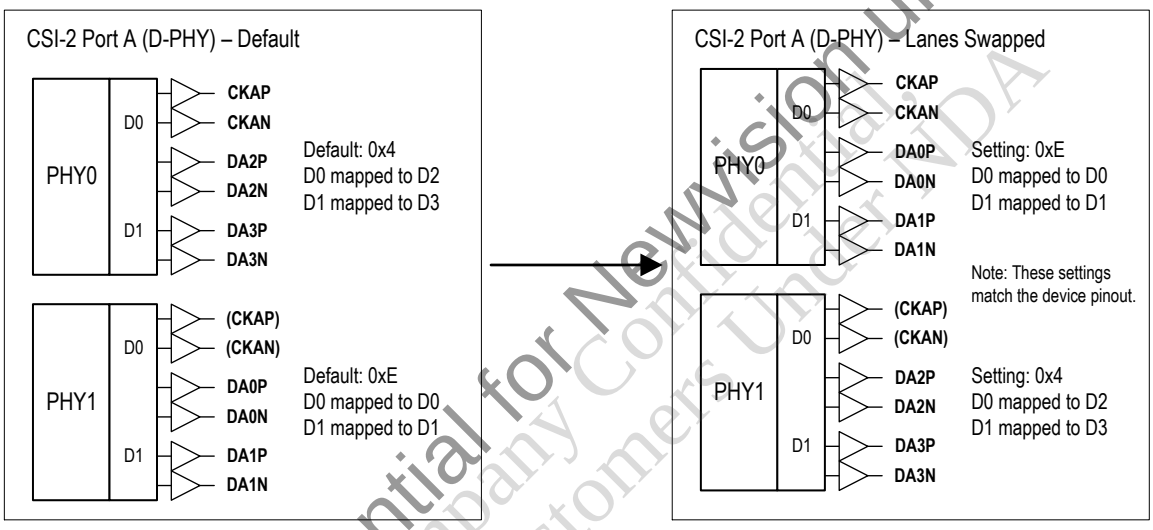
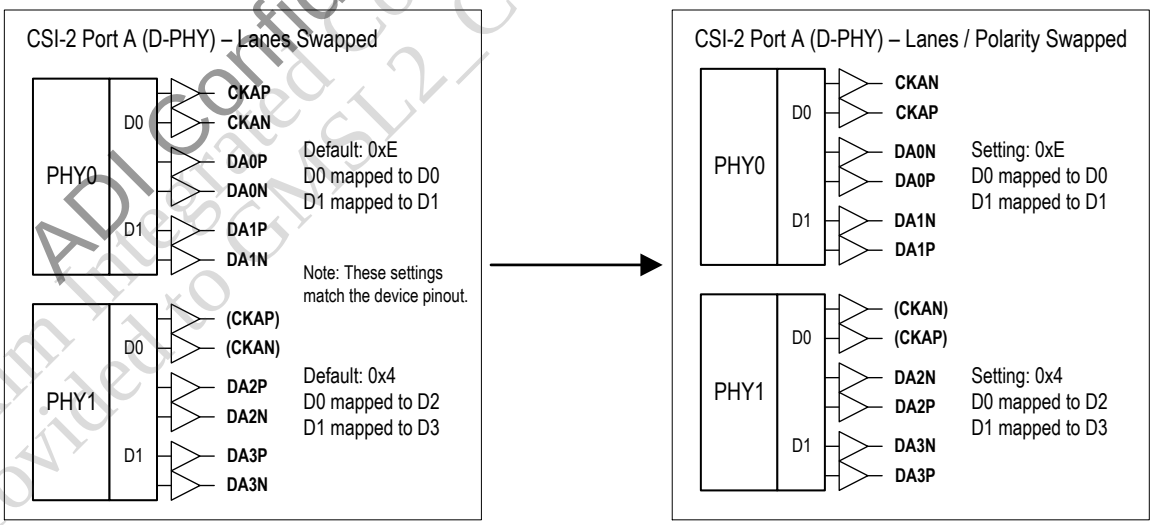


Figure 214. D-PHY Lane Swap Example



ADI Confidential for New vision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

Figure 215. D-PHY Polarity Swap Example

44.7.4.1.2 C-PHY Lane Swap Example

The location of each trio may be swapped within the same PHY in C-PHY mode. The default for all three output pins (i.e., A, B, and C) can be swapped from the default pinout. The polarity inversion function in C-PHY mode swaps the A and C pins of a trio. Figure 216 shows the lane swap required to match the device pinout in C-PHY mode. Figure 217 demonstrates polarity swap in C-PHY mode.

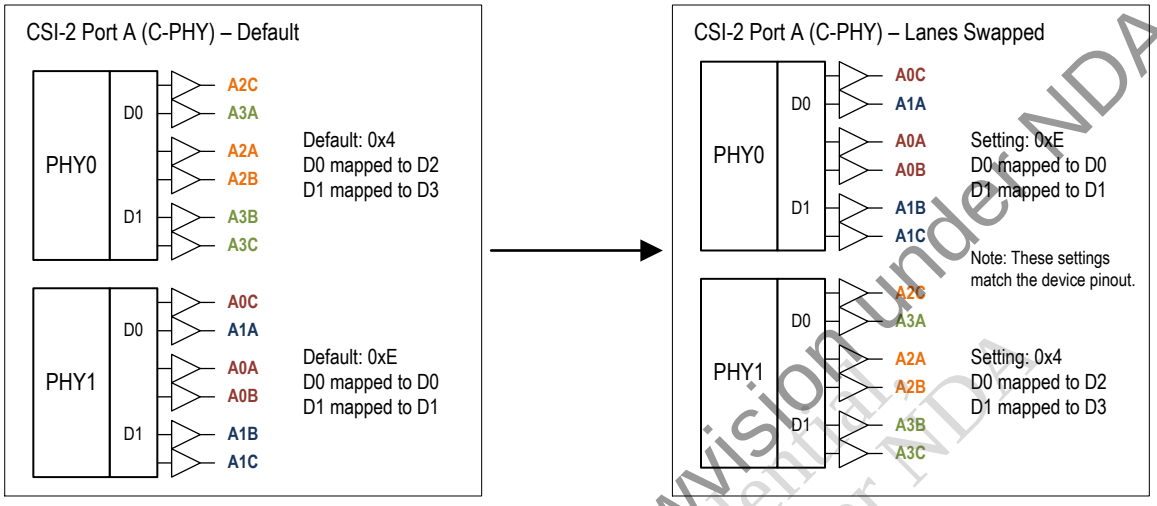


Figure 216. C-PHY Lane Swap Example

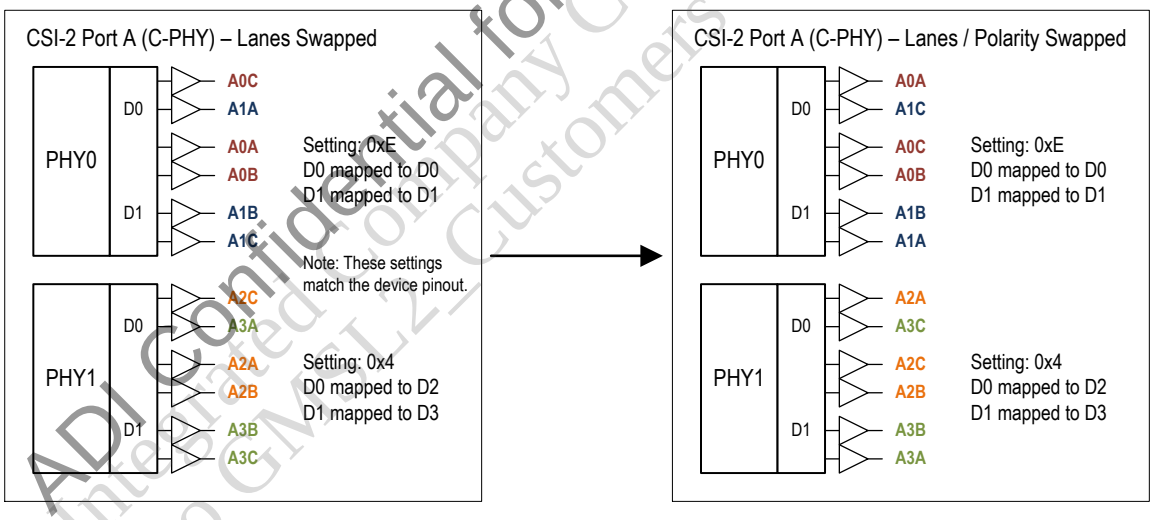


Figure 217. C-PHY Polarity Swap Example

Ar. ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.7.4.1.3 Programming Example

This example programs the MIPI PHY mode, configures the lane mappings, and sets the lane rates.

```
# Set MIPI PHY 2x4 mode
0x52,0x08A0,0x04
# Set lane mapping for ctrl 1 and 2 (MIPI Port A and B in x4 mode). This is written
to follow the device pinout.
0x52,0x08A3,0xE4
0x52,0x08A4,0xE4
# Set 4 lanes for ctrl 1 and 2 (MIPI Port A and B in x4 mode) in D-PHY mode
0x52,0x094A,0xC0
0x52,0x098A,0xC0
# Put DPLL for ctrl 1 and 2 in reset (config_soft_rst_n = 0) before changing MIPI
lane rates
0x52,0x1D00,0xF4
0x52,0x1E00,0xF4
# Set MIPI lane rate to be 2Gbps/lane for ctrl 1 and 2 (MIPI Port A and B in x4
mode)
0x52,0x0418,0x34
0x52,0x041B,0x34
# Release DPLL reset (config_soft_rst_n = 1)
0x52,0x1D00,0xF5
0x52,0x1E00,0xF5
# Turn on all MIPI PHYs (default)
0x52,0x08A2,0xF0
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.8 Use Case Examples

The following sections provide configuration details needed for common GMSL2 CSI-2 quad deserializer applications. These examples are included to serve as a reference of a known good configuration to modify for specific use cases.

- *Pairing with GMSL2 Serializers*
- *Bandwidth Efficiency Optimization*
- *Handling Multiple Datatypes*
- *Pairing with GMSL1 Serializers*
- *Software Override*
- *Video Data Replication*
- *Concatenation*
- *Frame Synchronization (FSYNC)*
- *Video Pattern Generator (VPG)*
- MIPI PRBS Generator
- GMSL2 Video PRBS Generator and Checker
- I2C/UART Control Channels
- I2C Broadcasting
- *Pass-Through I2C/UART*
- I2C Address Translation
- *Pass-Through SPI*
- *MIPI D-PHY Deskew Settings*
- *MIPI C-PHY Settings*
- *Sync Pulse Output*
- Video Mask Output
- Error-Correcting Code (ECC) Memory
- MIPI Packet Count

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.8.1 Pairing with GMSL2 Serializers

A common camera application for GMSL2 CSI-2 quad deserializers is pairing with multiple GMSL2 CSI-2 serializers. The programming steps described above can be applied to CSI-in/CSI-out system architecture.

This section provides further configuration details and includes instructions to program the serializer. Advanced techniques for more complicated setups are also introduced.

44.8.1.1 MAX9295A Programming

The MAX9295A is a single-port GMSL2 CSI-2 serializer. One or more MAX9295A devices can be paired with a single GMSL2 CSI-2 quad deserializer for camera applications. In GMSL2 mode, the serial link is automatically established when the serializer is connected to the CSI-2 quad deserializer (reference Link Initialization). After the link is established, the MAX9295A can be programmed. Important video path programming options include:

- MIPI PHY / MIPI Controller Settings
- Video Pipe Selection and Data Type filter Settings (if needed)

44.8.1.1.1 Programming Example

This example configures a MAX9295A so that video pipe X streams RAW12 data while pipe Y streams EMB8 data.

```
# MIPI 1x4 mode, 4-lane
0x80,0x0330,0x00
0x80,0x0331,0x33
0x80,0x0332,0xEE
0x80,0x0333,0xE4
# Enable video pipe X, Y from CSI Port B (used for 1x4 mode) and start video
transmitting
0x80,0x0308,0x63
0x80,0x0311,0x30
0x80,0x0002,0x33
# Enable data type filter RAW12 and emb8 for Pipe X and Y; MSB = enable, [6:0] = DT
0x80,0x0314,0x6C
0x80,0x0316,0x52
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.8.2 Bandwidth Efficiency Optimization

The 6Gbps GMSL2 forward link rate has an effective maximum video payload of approximately 5.2Gbps after accounting for the 9b/10b encoding and link protocol overhead. Similarly, the 3Gbps GMSL2 forward link rate has an approximate effective video bandwidth of 2.6Gbps. The incoming payload at serializer input must be below the relevant figure to ensure proper operation.

If an image sensor outputs MIPI CSI-2 data, the payload can be calculated by multiplying the lane rate by the number of lanes. For example, an image sensor outputting data with a lane rate of 450Mbps on four lanes:

$$Payload = 450Mbps * 4lanes = 1.8Gbps$$

Alternatively, image details may be used to calculate the payload. For example, a 2.3MP image sensor with a resolution of 1928 x 1208 running at 30fps outputs RAW12 datatype:

$$Payload \cong (1928 * 1208) * 1.3(blanking) * 30(fps) * 12(BPP) \cong 1.1Gbps$$

High-resolution cameras may require additional configuration to optimize bandwidth usage and ensure proper function of the serial link. For instance, an 8MP image sensor (RAW12) running at 30fps has a typical output of less than 4.5Gbps. This is within the data throughput limit of a GMSL2 6Gbps link, however, it will exceed the maximum allowed PCLK limit of 375MHz. Two settings (described below) provide configuration options for optimizing bandwidth efficiency.

The following equation is used to calculate the PCLK value:

$$PCLK = \frac{LANE_CNT * LANE_RATE}{bpp}$$

Refer to the Interface-Specific Bandwidth Calculations section for additional serial link bandwidth information.

44.8.2.1 Disabling "Heartbeat" Mode

Heartbeat mode is enabled by default. This mode is designed for GMSL display video links where HS, VS, and DE(HVD) values are transmitted using GMSL short packets during video blanking. The CSI-2 protocol uses its own long data packets for video data as well as FS/FE short packets for VS, so heartbeat mode is not necessary for CSI-2 data transmission and can be turned off to minimize serial link bandwidth usage.

Heartbeat mode is configured individually for each video pipe. It should only be disabled for pipes carrying image data; it should not be disabled if a pipe is only carrying embedded data, which usually appears as a few lines of each frame. A video pipe must continuously receive at least 16 valid video packets within a moving time window for VIDEO_LOCK to be asserted. This window becomes approximately 100 times larger when heartbeat mode is disabled.

To disable heartbeat mode:

- Set LIM_HEART = 1 in serializer.
- Set SEQ_MISS_EN = 0, DIS_PKT_DET = 0, and LIM_HEART = 1 in the MAX96712/722.

Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA
Analog Devices Confidential
ADI Confidential for Newvision under NDA

Note: The LIM_HEART bitfield has different functionality in the CSI-2 quad deserializer devices than GMSL2 serializers. In the MAX96712/722, this bitfield defines the timeout to detect loss of video lock.

44.8.2.1.1 Programming Example

This configuration disables heartbeat mode in both the serializer and in the quad deserializer.

```
# Efficiency updates for pipe X in MAX9295A that carries image data. LIM_HEART = 1
0x80,0x0102,0x0E
# Efficiency updates for pipe 0-3 in MAX96712/722 that carry image data.
SEQ_MISS_EN = 0
0x52,0x0100,0x22
0x52,0x0112,0x22
0x52,0x0124,0x22
0x52,0x0136,0x22
# Efficiency updates for pipe 0-3 in MAX96712/722 that carry image data. LIM_HEART
= 1
0x52,0x0106,0x0A
0x52,0x0118,0x0A
0x52,0x012A,0x0A
0x52,0x013C,0x0A
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.8.2.2 Using Double Mode

GMSL2 CSI-2 quad deserializers have a maximum allowed internal PCLK value of 375MHz when processing image data. For a MIPI CSI-2 input, the PCLK calculation for incoming video is the total payload divided by the bpp rate. For example, an 8MP RAW8 image:

$$PCLK = \frac{4.4Gbps}{8bpp} = 550MHz$$

This exceeds the maximum allowed value (375MHz). This can be addressed by enabling double pixel mode, which doubles the bpp to save bandwidth. GMSL protocol allocates 24 bits of each packet for video content. Therefore, double pixel mode can be used for any data type less than or equal to 12bpp. In the above example, doubling 8bpp would yield a 275MHz PCLK and meet the internal PCLK requirement for the GMSL2 CSI-2 quad deserializers. For more detailed examples of double mode and data type handling please refer to the CSI-2 section.

There are three bpp values that can be doubled: 8, 10, and 12bpp. The settings for each are listed below. All bpp configurations are individually programmed for each pipe with dedicated ALT_MEM_MAP registers. The * in the settings below should be replaced with the desired pipe names (i.e., X/Y/Z/U for serializers and 0–7 for GMSL2 CSI-2 quad deserializers).

1. Double pixel mode for 8bpp
   - Set bpp8dbl* = 1, soft_bpp*_en = 1, and soft_bpp* = 16 in the serializer.
   - Set bpp8dbl* = 1, bpp8dbl*_mode = 1, ALT_MEM_MAP8 = 1 in MAX96712/722.

2. Double pixel mode for 10bpp
   - Set bpp10dbl* = 1, soft_bpp*_en = 1, soft_bpp* = 20 in the serializer.
   - Set ALT_MEM_MAP10 = 1 in MAX96712/722.

3. Double pixel mode for 12bpp
   - Set bpp12dbl* = 1, soft_bpp*_en = 1, soft_bpp* = 24 in the serializer.
   - Set ALT_MEM_MAP12 = 1 In MAX96712/722.

Note: The operation for 8bpp double mode is different than 10/12bpp and requires two extra writes in the GMSL2 CSI-2 quad deserializer.

44.9 Refer to the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.

Analog Confidential - Restricted Distribution
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2_Customers Under NDA
ADI Confidential for New Vision under NDA

- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

44.9.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.

Maxim Integrated Confidential, Provided to GMSL2 Customers under NDA
ADI Confidential for new vision under NDA

Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock, lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

					training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

44.9.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_CUSTOMER Under NDA

44.10 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 section for additional information CSI-2 bandwidth efficiency optimization.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.1.1.1 Programming Example

This example shows the programming required to enable double mode in the serializer and connected deserializer.

```
# Enable bpp12/8 double mode for pipe X and Y in MAX9295A
# 24 bpp override on X and 16 bpp override on pipe Y
0x80,0x0312,0x02
0x80,0x0313,0x10
0x80,0x031C,0x38
0x80,0x031D,0x30
# Enable bpp12/8 on corresponding receive pipes in MAX96712
# bpp8dbl = 1 and bpp8dbl_mode = 1 for pipe 4-7
# ALT_MEM_MAP8/ALT_MEM_MAP12 = 1 on Ctrl 1 for port A output
0x52,0x0434,0xF0
0x52,0x0437,0xF0
0x52,0x0973,0x03
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.10.2 Handling Multiple Datatypes

Image frames containing more than one datatype can be combined into a single video pipe if the datatypes share the same frame rate. However, it is recommended that they be separated to maximize bandwidth usage. This is common with image frames that contain embedded lines. The GMSL2 CSI-2 quad deserializers have eight video pipes designed to support cases where each of the four connected serializers is streaming two data types (e.g., video data and embedded data).

Video pipes generate frame start (FS) and frame end (FE) short packets when streaming video. If two video pipes are carrying two different datatypes from the same image, only one of the FS/FE pairs is needed. Generating the extra FS/FE pair without mapping will result in a line buffer overflow.

44.10.2.1.1 Programming Example

The programming example describes an application where the GMSL2 CSI-2 quad deserializer uses two different video pipes for an incoming video that contains both RAW12 and EMB8 data in each frame.

```
# Disable the FS/FE packets output from RAW12 pipes and use the ones from EMB8
pipes
0x52,0x0436,0x0F
# Video Pipe to MIPI Controller Mapping
# RAW12 in video pipe 0, map only video data
0x52,0x090B,0x01
0x52,0x092D,0x01
0x52,0x090D,0x2C
0x52,0x090E,0x2C
# Embedded 8 in video pipe 4, map FS/FE
0x52,0x0A0B,0x07
0x52,0x0A2D,0x15
0x52,0x0A0D,0x12
0x52,0x0A0E,0x12
0x52,0x0A0F,0x00
0x52,0x0A10,0x00
0x52,0x0A11,0x01
0x52,0x0A12,0x01
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distributed Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.3 Pairing with GMSL1 Serializers

GMSL2 CSI-2 quad deserializers are GMSL1 backwards compatible and can be paired with GMSL1 serializers. Unlike GMSL2, the GMSL1 link does not automatically establish, and the forward video channel requires an external video pixel clock at the serializer input.

The following procedure is used to establish a GMSL1 link:

- Build the reverse control channel (the CLINK). There are two methods:
  - CFG pins settings (reference device datasheet CFG pin information).
  - Register configuration (see *Pairing with a HIM-Capable GMSL1 Serializer* below).
- Program the serializer and image sensor via the reverse channel.
- Program the GMSL2 CSI-2 quad deserializer following the steps described above in the Configuration section.
- Enable image sensor streaming and enable the forward channel.

When using the GMSL2 CSI-2 quad deserializer in GMSL1 mode, use GMSL1 camera serializers that are equipped with high immunity mode (HIM). HIM provides robust control channel EMC tolerance, which reduces the effects of bit errors and the potential to corrupt reverse channel communication. Devices operating without HIM can fail Bulk Current Injection (BCI) tests and should not be used in applications requiring Power over Coax (PoC).

Other GMSL1 legacy mode serializers without HIM support (e.g., MAX9271) can be used, but require additional design considerations. Building the CLINK without HIM enabled requires more configuration and is sensitive to hardware setup if PoC is used. Reference the **GMSL2 Hardware Design Guide** for backwards compatible PoC design solutions if the CSI-2 quad deserializer is to support both GMSL1 and GMSL2 applications.

A robust reverse communication channel is important when pairing the GMSL2 CSI-2 quad deserializer with GMSL1 devices. Enable packet-based control channel mode by setting the deserializer PKTCC_EN bits high for each link. If there is no I2C master on the remote side of the link, set NO_REM_MST high to reduce unnecessary timeouts on the communication channels.

Refer to the GMSL1 Legacy Mode (Backward Compatibility) section for additional information.

The following sub-sections provide detailed discussion about building the CLINK with GMSL1 serializers:

- Pairing with a HIM-Capable GMSL1 Serializer
- Pairing with a non-HIM-Capable GMSL1 Serializer

ADI Confidential for New Version under NDA
Analog Devices Confidential Property
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

44.10.3.1 Pairing with a HIM-Capable GMSL1 Serializer

A GMSL1 serializer with HIM mode enabled upon power-up (e.g., MAX96705A) is recommended for GMSL1 camera applications with the GMSL2 CSI-2 quad deserializers. This is achieved by adding a pull-up resistor at the HIM pin of the serializer. Refer to the MAX96705A datasheet for additional details.

There are two methods to build the CLINK. Power up the GMSL2 CSI-2 quad deserializer with GMSL1 HIM mode enabled by setting the CFG pins or enable HIM mode with register write HIGHHIMM (bit 7 in register 0x0B06/0x0C06/0x0D06/0x0E06 for link A/B/C/D respectively).

If a MAX96705A is powered up without HIM enabled, the CLINK must be built under non-HIM mode. After the CLINK is built, HIM mode can then be enabled through register writes.

The procedure is listed below (Figure 218 and Table 239). Note that before HIM mode is established on both sides, it is recommended to individually enable and program each link to avoid potential I2C command race conditions. This can be done by only having one link enabled on the deserializer at a time.

44.10.3.1.1 Programming Example

This example demonstrates how to set up the communication channel for a GMSL 1 link. Note that it is important to enable and program each link individually.

```
# Turn on HIM on MAX96712/722 Link A
0x52,0x0B06,0xEF
# Turn on local I2C acknowledge as link is not established yet
0x52,0x0B0D,0x80
# Enable CLINK in MAX96705A that is connected on Link A
0x80,0x0004,0x43
# Delay 5ms
# Turn off local I2C acknowledge
0x52,0x0B0D,0x00
```

ADI Confidential for New vision under NDA
Analog Devices Confidential, Distribute to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.10.3.1.2 Programming Example

This example shows how to set up a complete GMSL1 link and the required registers for the deserializer. This example assumes that the GMSL1 serializer is in HIM mode at power up.

```
# Enable Link A only
0x04, 0x52, 0x00, 0x06, 0x01,
# Set deserializer link A: HIM Mode Enable
0x04, 0x52, 0x0B, 0x06, 0xEF,
# Turn on local I2C acknowledge as link is not established yet
0x04, 0x52, 0x0B, 0x0D, 0x80,
# At this point you should have access to the serializer I2C communication. In GUI
you need to hit identify devices under options tab.

# Turn on CLINK link and turn video link off.
0x03, 0x80, 0x04, 0x43,

# Serializer Enable double and HVEN
0x03, 0x80, 0x07, 0x84,
# Deserializer Enable double and HVEN
0x04, 0x52, 0x0B, 0x07, 0x84,
# Deserializer Disable Remote master
0x04, 0x52, 0x0B, 0x05, 0x79,
# Deserializer Enable SHIFT_VID_HVD
0x04, 0x52, 0x0B, 0xA7, 0x45,
# Disable I2C_LOC_ACK: This should be disabled once the forward channel is
established.
0x04, 0x52, 0x0B, 0x0D, 0x00,
# Deserializer, Set datatype to YUV422. This can be changed depending on the data
type used.
0x04, 0x52, 0x0B, 0x96, 0x1B,
# Deserializer, DE_EN = 0; This is dependent on the camera used.
0x04, 0x52, 0x0B, 0x0F, 0x01,
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

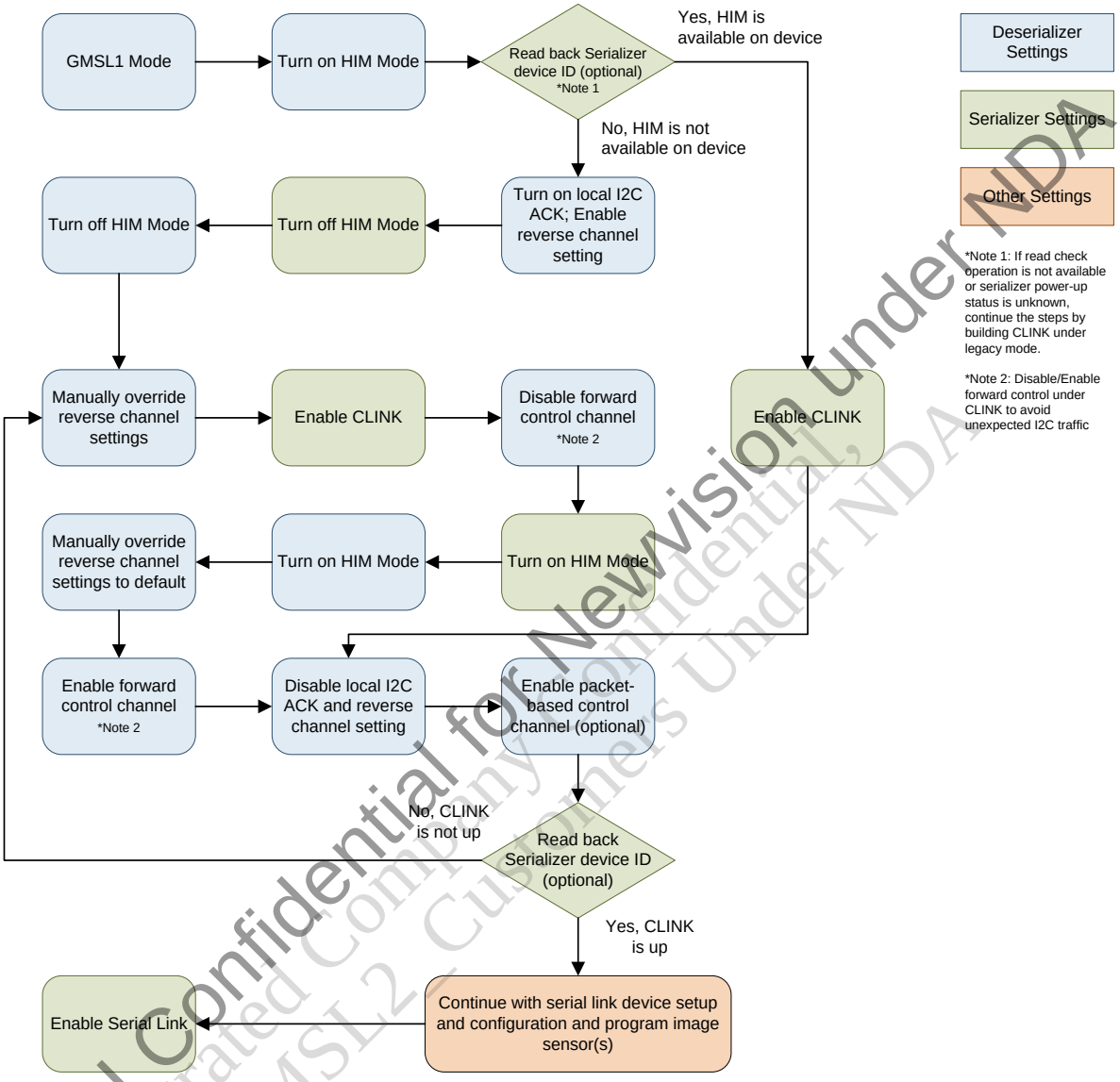


Figure 218. Building the CLINK on HIM-Capable GMSL1 Serializer

Note: The process detailed in Figure 218 and Table 239 applies to HIM-Capable GMSL1 serializers.

ADI Confidential for Newvision under NDA
Analog Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

Table 239. Configuration Steps for Pairing HIM-Capable GMSL1 Serializers

Step	Device	Slave ID	Register(s)	Value	Description
1	Quad DES	0x52	0x0006	0x01/ 0x02/ 0x04/ 0x08	Enable Link A/B/C/D in GMSL1 mode. Each link being programmed must be enabled.
Delay 5ms					
2	Quad DES	0x52	0x0B06/ 0x0C06/ 0x0D06/ 0x0E06	0xEF	Turn on HIM mode on Link A/B/C/D. *This step should be skipped if the serializer is not in HIM mode.
Note					If Ser is powered up with HIM by default, CLINK is established with one more write on Ser (0x04=0x43). User can proceed to step 17. Otherwise continue.
3	Quad DES	0x52	0x0B0D/ 0x0C0D/ 0x0D0D/ 0x0E0D	0x81	Enable reverse channel config on Link A/B/C/D; Turn on local I2C acknowledge on Link A/B/C/D.
4	SER	0x80	0x004D	0x40	Turn off HIM on Ser.
Delay 10ms					
5	Quad DES	0x52	0x0B06/ 0x0C06/ 0x0D06/ 0x0E06	0x6F	Turn off HIM mode on Link A/B/C/D.
6	Quad DES	0x52	0x14C5/ 0x15C5/ 0x16C5/ 0x17C5	0xAA	Manually override reverse channel pulse length for Link A/B/C/D.
7	Quad DES	0x52	0x14C4/ 0x15C4/ 0x16C4/ 0x17C4	0x80	Manually override reverse channel rise/fall time for Link A/B/C/D.
8	Quad DES	0x52	0x1495/ 0x1595/ 0x1695/ 0x1795	0xC8	Manually override reverse channel Tx amplitude for Link A/B/C/D.
9	SER	0x80	0x0004	0x43	Enable control link only; Disable serial link.
Delay 5ms					Control link should be built up successfully.
10	Quad DES	0x52	0x0B04/ 0x0C04/ 0x0D04/ 0x0E04	0x02	Disable forward control channel transmitter for Link A/B/C/D.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

11	SER	0x80	0x004D	0xC0	Enable HIM mode on Ser (i.e., remote side).
Delay 5ms					
12	Quad DES	0x52	0x0B06/ 0x0C06/ 0x0D06/ 0x0E06	0xEF	Turn on HIM mode on Link A/B/C/D.
Repeat above steps for all other links. **CLINK should be established on all links with HIM enabled.**					
13	Quad DES	0x52	0x14C5/ 0x15C5/ 0x16C5/ 0x17C5	0x40	Manually override reverse channel pulse length to default value for Link A/B/C/D.
14	Quad DES	0x52	0x14C4/ 0x15C4/ 0x16C4/ 0x17C4	0x40	Manually override reverse channel rise/fall time to default value for Link A/B/C/D.
15	Quad DES	0x52	0x1495/ 0x1595/ 0x1695/ 0x1795	0x69	Manually override reverse channel Tx amplitude to default value for Link A/B/C/D.
16	Quad DES	0x52	0x0B04/ 0x0C04/ 0x0D04/ 0x0E04	0x03	Enable forward control channel transmitter for Link A/B/C/D.
17	Quad DES	0x52	0x0B0D/ 0x0C0D/ 0x0D0D/ 0x0E0D	0x00	Disable local I2C acknowledge; Disable reverse channel setting.
18 (optional)	Quad DES	0x52	0x0B08	0x25	Enable packet-based control channel mode for Link A.
Delay 10ms					
			0x0C08	0x25	Enable packet-based control channel mode for Link B.
Delay 10ms					
			0x0D08	0x25	Enable packet-based control channel mode for Link C.
Delay 10ms					
			0x0E08	0x25	Enable packet-based control channel mode for Link D.
Delay 10ms					

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Subject to NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Continue programming the Quad Deserializer.					
Continue programming the Serializer and Camera.					
Delay 10ms					
19	SER	0x80	0x0004	0x83	Enable serial link.
Last	DES	0x52	0x0006	0x0F	Enable all four links (A, B, C, and D).

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.10.3.2 Pairing with a non-HIM-Capable GMSL1 Serializer

Non-HIM-capable serializers (e.g., MAX9271) should be replaced with HIM-capable serializers (e.g., MAX96705A) for new GMSL1 camera link designs. If a non-HIM-capable GMSL1 serializers must be used, the procedure below details building a reliable reverse control channel (Figure 219 and Table 240).

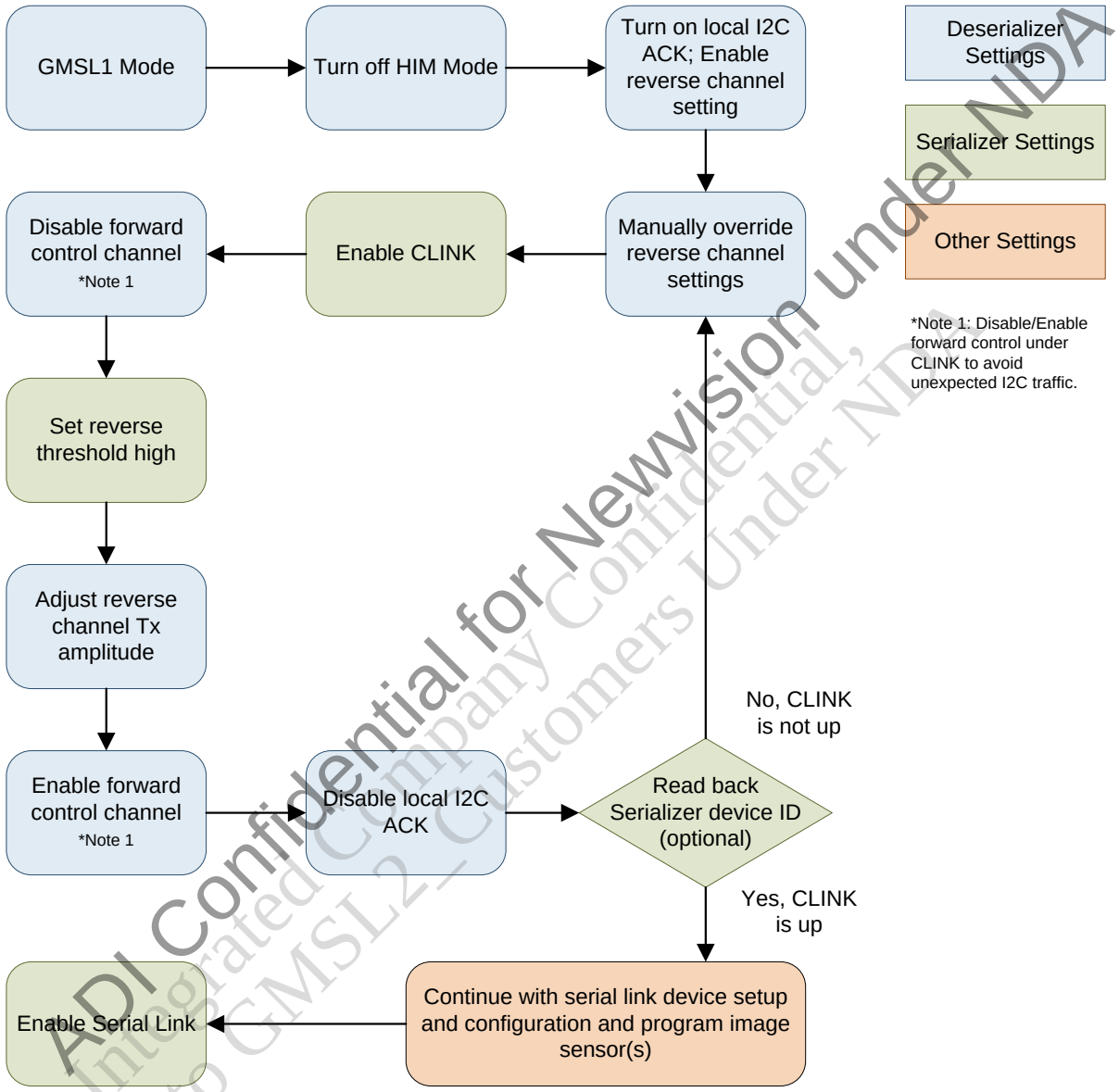


Figure 219. Building the CLINK on non-HIM-Capable GMSL1 Serializer

Note: The process detailed in Figure 219 and Table 240 applies to non-HIM-Capable GMSL1 serializers.

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

Table 240. Configuration Steps for Pairing non-HIM-Capable GMSL1 Serializers

Step	Device	Slave ID	Register(s)	Value	Description
1	Quad DES	0x52	0x0006	0x01/ 0x02/ 0x04/ 0x08	Enable Link A/B/C/D in GMSL1 mode.
Delay 5ms					
2	Quad DES	0x52	0x0B06/ 0x0C06/ 0x0D06/ 0x0E06	0x6F	Turn off HIM mode on Link A/B/C/D.
3	Quad DES	0x52	0x0B0D/ 0x0C0D/ 0x0D0D/ 0x0E0D	0x81	Enable reverse channel config on Link A/B/C/D; Turn on local I2C acknowledge on Link A/B/C/D.
4	Quad DES	0x52	0x14C5/ 0x15C5/ 0x16C5/ 0x17C5	0xAA	Manually override reverse channel pulse length for Link A/B/C/D.
5	Quad DES	0x52	0x14C4/ 0x15C4/ 0x16C4/ 0x17C4	0x80	Manually override reverse channel rise/fall time for Link A/B/C/D.
6	Quad DES	0x52	0x1495/ 0x1595/ 0x1695/ 0x1795	0xC8	Manually override reverse channel Tx amplitude for Link A/B/C/D.
7	SER	0x80	0x04	0x43	Enable control link only; Disable serial link.
Delay 10ms					
8	Quad DES	0x52	0x0B04/ 0x0C04/ 0x0D04/ 0x0E04	0x02	Disable forward control channel transmitter for Link A/B/C/D.
Delay 5ms					
9	SER	0x80	0x08	0x08	Set threshold high in Ser to comply with Des reverse channel settings; REV_HIVTH=1, REV_GAIN=0.
Delay 5ms					
10	Quad DES	0x52	0x1495/ 0X1595/ 0X1695/ 0X1795	0xD3	Manually increase reverse channel Tx amplitude for Link A/B/C/D.
11	Quad DES	0x52	0x0B04/ 0x0C04/ 0x0D04/ 0x0E04	0x03	Enable forward control channel transmitter for Link A/B/C/D.

ADI Confidential for New Vision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

12	Quad DES	0x52	0x0B0D/ 0x0C0D/ 0x0D0D/ 0x0E0D	0x01	Disable local I2C acknowledge on Link A/B/C/D.
Repeat above steps for all other links. **CLINK should be established on all links under non-HIM mode.**					
Continue programming the Quad Deserializer.					
Continue programming the Serializer and Camera.					
Delay 10ms					
Last	SER	0x80	0x04	0x83	Enable serial link.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only to Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.10.4 Software Override

The software override is used to manually override the video data type (DT) (i.e., packet header), virtual channel number (VC), or bits per pixel (bpp). This operation affects the specification of the video data between the video pipe(s) and the MIPI controller(s). Overriding the DT and VC information is used for MIPI controller mapping. If the received video data is from a serializer in parallel mode (e.g., GMSL1 serializers), it is necessary to specify desired DT, VC, and bpp with the software override. See Table 241 for a list of software override registers.

Note: VC can be changed individually, however DT and bpp must be adjusted together to ensure settings compatibility.

- DT: soft_dt_x[5:0] E.g., RGB888: DT = 0x24 = 0b100100
- VC: soft_vc_x[3:0] E.g., VC: 3 = 0x03 = 0b0011
- bpp: soft_bpp_x[4:0] E.g., RAW12: bpp = 0xC = 0b01100

Table 241. Software Override Registers

Register	Bits	Default Value	Description
0x040B	7:3	00000	**Pipe 0 BPP Software Override Register:** Software override video data BPP.
0x040C	7:0	0x00	**Pipe 1 and 0 VC Software Override Register:** Bits [7:4]: Pipe 1 VC. Bits [3:0]: Pipe 0 VC.
0x040D	7:0	0x00	**Pipe 3 and 2 VC Software Override Register:** Bits [7:4]: Pipe 3 VC. Bits [3:0]: Pipe 2 VC.
0x040E	7:0	0x00	**Pipe 1 (H) and 0 DT Software Override Register:** Bits [7:6]: Pipe 1 DT (High Bits [5:4]). Bits [5:0]: Pipe 0 DT.
0x040F	7:0	0x00	**Pipe 2 (H) and 1 (L) DT Software Override Register:** Bits [7:4]: Pipe 2 DT (High Bits [5:2]). Bits [3:0]: Pipe 1 DT (Low Bits [3:0]).
0x0410	7:0	0x00	**Pipe 3 and 2 (L) DT Software Override Register:** Bits [7:2]: Pipe 3 DT. Bits [1:0]: Pipe 2 DT (Low Bits [1:0]).
0x0411	7:0	0x00	**Pipe 2 (H) and Pipe 1 BPP Software Override Register:** Bits [7:5]: Pipe 2 BPP (High Bits [4:2]). Bits [4:0]: Pipe 1 BPP.
0x0412	7:0	0x00	**Pipe 3 and Pipe 2 (L) BPP Software Override Register:** Bit 7: Reserved. Bits [6:2]: Pipe 3 BPP. Bits [1:0]: Pipe 2 BPP (Low Bits [1:0]).
0x0415	7:6	0x00	**Pipe 1 and 0 Software Override Enable Register:** Bit 7: Pipe 1. Bit 6: Pipe 0. 0 = Disable, 1 = Enable.
0x0418	7:6	0x00	**Pipe 3 and 2 Software Override Enable Register:** Bit 7: Pipe 3. Bit 6: Pipe 2. 0 = Disable, 1 = Enable.

ADI Confidential for Review under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customer under NDA

0x041B	7:6	0x00	**Pipe 5 and 4 Software Override Enable Register:** Bit 7: Pipe 5. Bit 6: Pipe 4. 0 = Disable, 1 = Enable.
0x041D	7:6	0x00	**Pipe 7 and 6 Software Override Enable Register:** Bit 7: Pipe 7. Bit 6: Pipe 6. 0 = Disable, 1 = Enable.
0x042B	7:3	0x00000	**Pipe 4 BPP Software Override Register:** Software override video data BPP.
0x042C	7:0	0x00	**Pipe 5 and 4 VC Software Override Register:** Bits [7:4]: Pipe 5 VC. Bits [3:0]: Pipe 4 VC.
0x042D	7:0	0x00	**Pipe 7 and 6 VC Software Override Register:** Bits [7:4]: Pipe 7 VC. Bits [3:0]: Pipe 6 VC.
0x042E	7:0	0x00	**Pipe 5 (H) and 4 DT Software Override Register:** Bits [7:6]: Pipe 5 DT (High Bits [5:4]). Bits [5:0]: Pipe 4 DT.
0x042F	7:0	0x00	**Pipe 6 (H) and 5 (L) DT Software Override Register:** Bits [7:4]: Pipe 6 DT (High Bits [5:2]). Bits [3:0]: Pipe 5 DT (Low Bits [3:0]).
0x0430	7:0	0x00	**Pipe 7 and 6 (L) DT Software Override Register:** Bits [7:2]: Pipe 7 DT. Bits [1:0]: Pipe 6 DT (Low Bits [1:0]).
0x0431	7:0	0x00	**Pipe 6 (H) and Pipe 5 BPP Software Override Register:** Bits [7:5]: Pipe 6 BPP (High Bits [4:2]). Bits [4:0]: Pipe 5 BPP.
0x0432	7:0	0x00	**Pipe 7 and Pipe 6 (L) BPP Software Override Register:** Bit 7: Reserved. Bits [6:2]: Pipe 7 BPP. Bits [1:0]: Pipe 6 BPP (Low Bits [1:0]).

44.10.4.1.1 Programming Example

The following script performs a series of VC, DT, and bpp software overrides on video pipes 0, 1, 2, and 3.

```
# VC/DT/BPP Software override for Pipe 0, 1, 2, and 3.
# 12 BPP override for pipe 0
0x52,0x040B,0x62
# VC 0 override for pipe 0/1/2/3
0x52,0x040C,0x00
0x52,0x040D,0x00
# RAW12 (0x2C) DT override for pipe 1/2/3/4
0x52,0x040E,0xAC
0x52,0x040F,0xBC
0x52,0x0410,0xB0
# 12 BPP override for pipe 2/3/4
0x52,0x0411,0x6C
0x52,0x0412,0x30
# Enable the software override for 1/2/3/4; Data rate = 600Mbps/lane
0x52,0x0415,0xE6
0x52,0x0418,0xE6
```

Maxim Integrated Confidential
ADI Confidential for Viewing Only
Provided to Company Confidential Under NDA
Maxim Integrated Confidential
ADI Confidential for Viewing Only
Provided to Company Confidential Under NDA
Maxim Integrated Confidential
ADI Confidential for Viewing Only
Provided to GMSL2 Customer Under NDA

44.10.5 Video Data Replication

Video data replication is used to mirror video streams. GMSL2 CSI-2 quad deserializers provide two methods to replicate video.

44.10.5.1 Video Replication at the Video Pipes

This method allows video data from a single source to be streamed from multiple video pipes. This is enabled by selecting the same video stream ID at the desired video pipes. Reference *Video Pipe Selection* for register details. The MIPI PHY controller video output destination selection follows the procedures discussed in *Video Pipe to MIPI Controller Mapping*.

Note: An unused video pipe must be available to be configured as a destination for video replication.

44.10.5.2 Video Replication at the MIPI PHYs

GMSL2 CSI-2 quad deserializers also support MIPI PHY-to-PHY replication. This is configured by enabling PHY copy and setting the source and destination PHYs with phy_cp0[4:0] and phy_cp1[4:0] in registers 0x08A9 and 0x08AA, respectively. Table 242 contains MIPI PHY replication registers.

Note: Two replications can be enabled at the same time in x2 mode (see MIPI PHY Settings for details).

Table 242. MIPI PHY Replication Registers

Register	Bits	Default Value	Description
0x08A9	7:3	00000	**PHY copy Register:** Bit 7: 0 = Disable, 1 = Enable PHY0 copy Bits [6:5]: PHY copy destination. Bits [4:3]: PHY copy source.
0x08AA	7:3	00000	**PHY copy Register:** Bit 7: 0 = Disable, 1 = Enable PHY1 copy Bits [6:5]: PHY copy destination. Bits [4:3]: PHY copy source.

44.10.5.2.1 Programming Example

This example performs link replication on Link A, pipe Z to pipes 0-3. The video from the serializer connected on Link A is replicated four times.

```
# Selects Link A, pipe Z, for pipe 0 and pipe 1.
0x52, 0xF0, 0x22
# Selects Link A, pipe Z, for pipe 2 and pipe 3.
0x52, 0xF1, 0x22
# Only turn on pipes 0-3
0x52, 0xF4, 0x0F
```

ADI Confidential for new version under NDA
Maxim Integrated Confidential, Provided to GMSL_ Customers under NDA

44.10.6 Concatenation

Typically, video data from multiple video pipes can be aggregated via mapping to one MIPI port through a virtual channel (VC) on a first-come, first-served (FCFS) basis. However, in some cases the SoC cannot parse VCs, and it is recommended to combine multiple streams into a single superframe (Figure 220). The SoC can then manipulate the data and separate the images if needed. This operation requires that pipe-to-controller mapping be disabled and that video sources be frame-synchronized. In this mode, all video streams share a single VC. This is also referred to as synchronous aggregation. Table 243 contains frame concatenation registers.

Note: Concatenation is only available to video pipes 0-3.

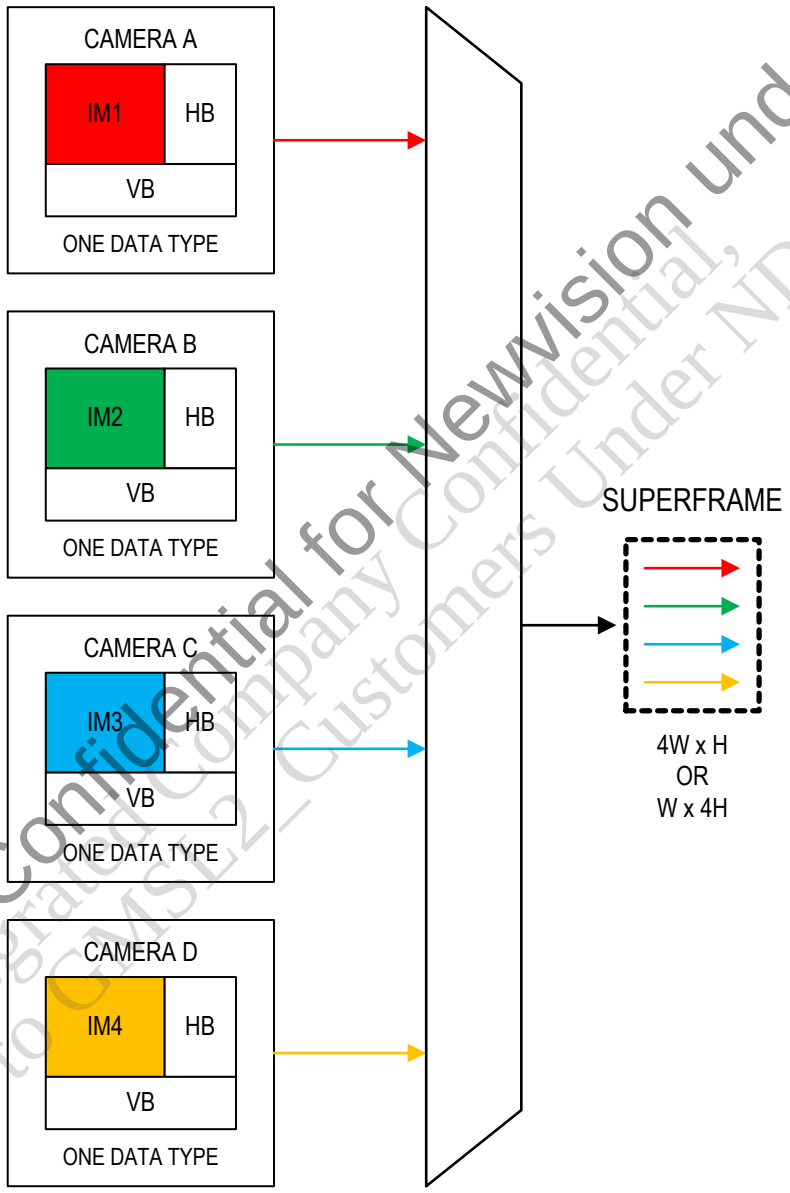


Figure 220. Video Pipe Input and Concatenated Superframe Output

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

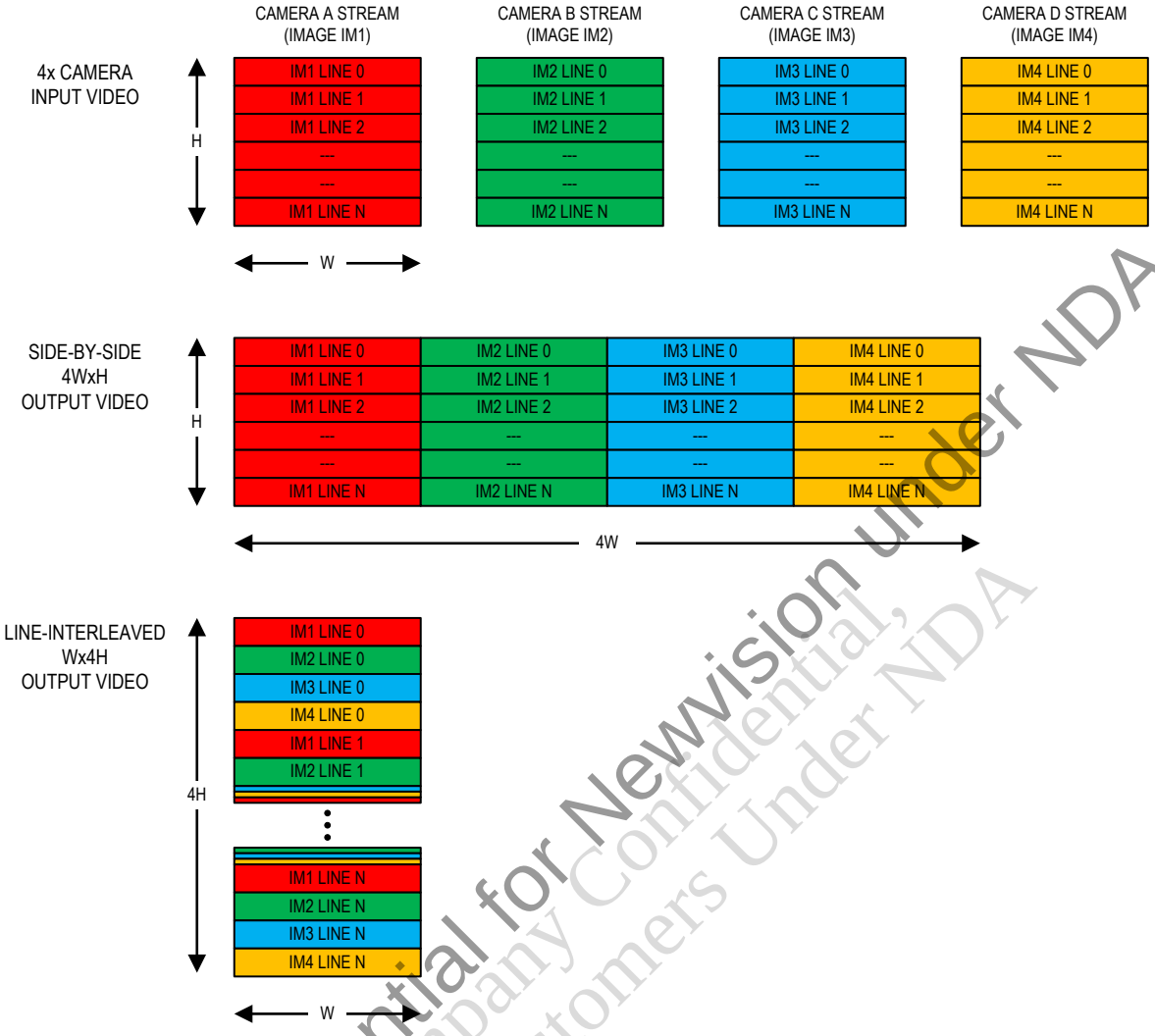


Figure 221. Concatenation (4WxH and Wx4H)

GMSL2 CSI-2 quad deserializers support either 4WxH (side-by-side) or Wx4H (line-interleaved) concatenation (Figure 221). Note that concatenation can be used with fewer than four streams. The superframe aggregated output is shown in Figure 222.

- **4WxH mode**: image pixel data is concatenated as a super line; number of lines remains the same.
- **Wx4H mode**: pixel resolution remains the same; lines are concatenated.

ADI Confidential for Newvision under NDA
Analog Devices Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

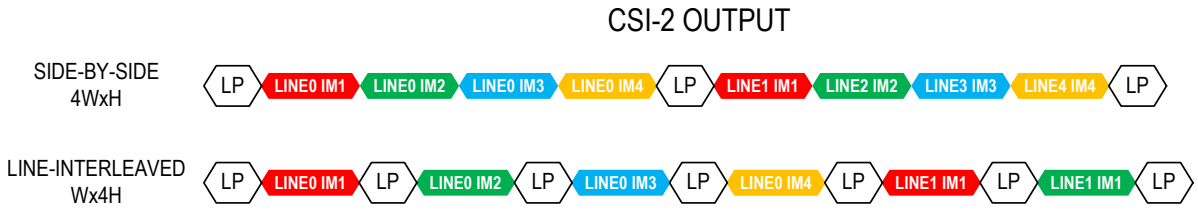


Figure 222. Superframe Aggregation

In 4WxH mode, the expected number of pixels for each stream is based on the bpp rate:

- If bpp ≤ 16, the number of pixels in a line must be a multiple of 8.
- If bpp > 16, the number of pixels in a line must be a multiple of 4.

If the above criteria are not met, black pixels will fill the remainder. For example, if four RAW12 (bpp = 12) camera streams (1026x800 each) are combined into a 4WxH superframe, the total number of pixels in a line is expected to be 4128 instead of 4104. This is because 1026 is not divisible by 8, and the closest available multiple of 8 is 1032.

In either in FCFS or concatenation mode, the MIPI output will continue if a link loses lock and the LOCK pin is de-asserted. However, the MIPI output stops if all four links lose lock.

Table 243. Frame Concatenation Registers

Register	Bits	Default Value	Description
0x0931	7:0	0x00	**MIPI Controller 0 Concatenation Register:** Bit 7: 0 = Wx4H mode, 1 = 4WxH mode. Bit 6: Reserved. Bits [5:4]: Select for the first line to concatenate; All others follow in order bit [3:0]. 00 = Select pipe 0 to be the master 01 = Select pipe 1 to be the master 10 = Select pipe 2 to be the master 11 = Select pipe 3 to be the master Bit 3: 0 = disable, 1 = concatenate video pipe 3. Bit 2: 0 = disable, 1 = concatenate video pipe 2. Bit 1: 0 = disable, 1 = concatenate video pipe 1. Bit 0: 0 = disable, 1 = concatenate video pipe 0.
0x0971	7:0	0x00	**MIPI Controller 1 Concatenation Register:** Bit 7: 0 = Wx4H mode, 1 = 4WxH mode. Bit 6: Reserved. Bits [5:4]: Select for the first line to concatenate; All others follow in order bit [3:0]. 00 = Select pipe 0 to be the master 01 = Select pipe 1 to be the master 10 = Select pipe 2 to be the master 11 = Select pipe 3 to be the master Bit 3: 0 = disable, 1 = concatenate video pipe 3. Bit 2: 0 = disable, 1 = concatenate video pipe 2. Bit 1: 0 = disable, 1 = concatenate video pipe 1.

ADI Confidential for New Vision Under NDA
Analog Devices Confidential, Distribution Only Under NDA
Maxim Integrated Company Confidential, Distribution Only Under NDA
Provided to GMSL2_Customers Under NDA

			Bit 0: 0 = disable, 1 = concatenate video pipe 0.
0x09B1	7:0	0x00	**MIPI Controller 2 Concatenation Register:** Bit 7: 0 = Wx4H mode, 1 = 4WxH mode. Bit 6: Reserved. Bits [5:4]: Select for the first line to concatenate; All others follow in order bit [3:0]. 00 = Select pipe 0 to be the master 01 = Select pipe 1 to be the master 10 = Select pipe 2 to be the master 11 = Select pipe 3 to be the master Bit 3: 0 = disable, 1 = concatenate video pipe 3. Bit 2: 0 = disable, 1 = concatenate video pipe 2. Bit 1: 0 = disable, 1 = concatenate video pipe 1. Bit 0: 0 = disable, 1 = concatenate video pipe 0.
0x09F1	7:0	0x00	**MIPI Controller 3 Concatenation Register:** Bit 7: 0 = Wx4H mode, 1 = 4WxH mode. Bit 6: Reserved. Bits [5:4]: Select for the first line to concatenate; All others follow in order bit [3:0]. 00 = Select pipe 0 to be the master 01 = Select pipe 1 to be the master 10 = Select pipe 2 to be the master 11 = Select pipe 3 to be the master Bit 3: 0 = disable, 1 = concatenate video pipe 3. Bit 2: 0 = disable, 1 = concatenate video pipe 2. Bit 1: 0 = disable, 1 = concatenate video pipe 1. Bit 0: 0 = disable, 1 = concatenate video pipe 0.

44.10.6.1.1 Programming Example

The following example configures 4WxH concatenation on Controller 1 for port A MIPI output in 2x4 mode.

```
# Enable 4WxH on Controller 1 for port A MIPI output in 2x4 mode
# All pipes are enabled and pipe 1 is the master
0x52,0x0971,0x9F
```

ADI Confidential for review under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.10.7 Frame Synchronization (FSYNC)

Frame synchronization (FSYNC) is used to align images sent from multiple sources in surround-view applications and is required for *Concatenation*. In FSYNC mode, the GMSL2 CSI-2 quad deserializer sends a sync signal to each serializer; the serializers then send the signal to the connected sensor.

There are two types of FSYNC methods available on the GMSL2 CSI-2 quad deserializers: internal frame sync and external frame sync. With internal frame sync, the GMSL2 CSI-2 quad deserializer generates the sync signal internally. The sync signal frequency and period must be specified in terms of the onboard crystal clock (25MHz). With external frame sync, the GMSL2 CSI-2 quad deserializer forwards a sync signal generated by a SoC. In GMSL2 application, any of the serializer or deserializer GPIOs can be used as a sync signal input/output by using GPIO forwarding. However, in GMSL1 mode, there are only specific GPI/GPO function pins that are available for sync signal forwarding (refer to device datasheet for more information).

In FSYNC mode, the auto-masking feature is enabled by default to mask corrupted images with blank screens. When enabled, the overall output is not impacted if any of the links are dropped.

Table 244 contains frame sync registers. Reference the Frame Sync section for additional details.

Note: External frame sync is recommended if the SoC can generate a sync signal.

Table 244. Frame Sync Registers

Register	Bits	Default Value	Description
0x04A0	7:0	0x00	**Frame Sync (FSYNC) Setting Register:** Bit 7: When enabled, memory overflow resets frame sync generation. Bit 6: Select FSYNC falling transition. 0 = Set at the middle of frame. 1 = Set immediately after rising transition. Bit 5: Select GPIO to output FSYNC signal (only works when FSYNC_MODE = 01). 0 = MFP2, 1 = MFP10 Bit 4: EN_VS_GEN Keep it 0 (for most of cases) to use VS received from source. 0 = Use VS from source; VS is not generated internally. 1 = VS is generated internally disregard source VS. Bits [3:2]: FSYNC_MODE 00 = FSYNC on; GPIO is not used as FSYNC input / output; Set this mode for internal frame sync mode. 01 = FSYNC on; GPIO is used as FSYNC output and drives a slave device; Set this mode for the master deserializer when have multiple ones. 10 = FSYNC off; GPIO is used as FSYNC input driven by a master device; Set this mode for slave deserializer(s). 11 = FSYNC off; GPIO is not used as FSYNC input / output; Set this mode for external frame sync mode. Bits [1:0]: FSYNC Method. 00 = Manual 01 = Semi-auto 10 = Auto 11 = Reserved
0x04A2	7:5	100	**FSYNC Master Link Selection Register:**

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Distribution Limited
Provided to GMSL2 Customers Under NDA

			000 = Select pipe 0 to be the master 001 = Select pipe 1 to be the master 010 = Select pipe 2 to be the master 011 = Select pipe 3 to be the master 1xx = Auto Select
0x04A5	7:0	0x00	**FSYNC Period Low Register:** Set FSYNC period (Bits [7:0]) in terms of clock cycles. (Effective when FSYNC Method = 00 and FSYNC Mode = 0x.)
0x04A6	7:0	0x00	**FSYNC Period Middle Register:** Set FSYNC period (Bits [15:8]) in terms of clock cycles. (Effective when FSYNC Method = 00 and FSYNC Mode = 0x.)
0x04A7	7:0	0x00	**FSYNC Period High Register:** Set FSYNC period (Bits [23:16]) in terms of clock cycles. (Effective when FSYNC Method = 00 and FSYNC Mode = 0x.)
0x04AF	7:0	0x9F	**FSYNC Setting Register:** **Bit 7:** Select the type of FSYNC signal to output from GPIO. 0 = GMSL1 type, 1 = GMSL2 type. **Bit 6:** Uses crystal oscillator clock for generating frame sync signal. 0 = Disabled, 1 = Enabled. This bit should be enabled when generating FSYNC. **Bit 4:** Select how links are selected for FSYNC generation. 0 = Include links selected by FS_LINK_x registers. 1 = Include all enabled links. **Bit 3:** FS_LINK_3 in FSYNC generation. 0 = Do not include video pipe 3. 1 = Include video pipe. **Bit 2:** FS_LINK_2 in FSYNC generation. 0 = Do not include video pipe 2. 1 = Include video pipe. **Bit 1:** FS_LINK_1 in FSYNC generation. 0 = Do not include video pipe 1. 1 = Include video pipe. **Bit 0:** FS_LINK_0 in FSYNC generation. 0 = Do not include video pipe 0. 1 = Include video pipe.
0x04B0	7:0	0x00	**FSYNC Error Counter Register (read only):** Bits [7:0]: Report FSYNC error counter; Reset to 0 when read or when FSYNC_LOCKED asserted.
0x04B5	7:0	0x00	**FSYNC Difference Register (read only):** Bits [7:0]: Report the difference (Low Bits [7:0]) between the fastest and the slowest frame in terms of master PCLK cycles.
0x04B6	7:0	0xx00	**FSYNC Difference and Lock Register (read only):** Bit 7: FSYNC_LOSS_OF_LOCK 0 = FSYNC loss of lock not detected. 1 = FSYNC loss of lock detected. Bit 6: FSYNC_LOSS_LOCKED (works only for internal frame sync modes) 0 = FSYNC is not locked. 1 = FSYNC is locked. Bits [5:0]: Report the difference (High Bits [13:8]) between the fastest and the slowest frame in terms of master PCLK cycles.

ADI Confidential for review only under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

44.10.7.1.1 Programming Examples

44.10.7.1.1.1 Internal FSYNC (GMSL1)

This example demonstrates the programming sequence required to enable internal FSYNC in GMSL1 mode.

```
# Enable Internal FSYNC manual mode
0x52,0x04A0,0x04
# Turn off auto master link selection
0x52,0x04A2,0x00
# Disable overlap window (Default in rev D)
0x52,0x04AA,0x00
0x52,0x04AB,0x00
# Set to use 25MHz XTAL; AUTO_FS_LINKS = 0, FS_USE_XTAL = 1, FS_LINK_x[3:0] = 0,
GMSL1
0x52,0x04AF,0x40
# Set FSYNC period to 25M/30 clock cycles. CLK = 25MHz. Sync freq = 30Hz
0x52,0x04A7,0x0C
0x52,0x04A6,0xB7
0x52,0x04A5,0x35
# Enable FSYNC transmission to serializer. Select GPI_1 (MAX96712 MFP2)
0x52,0x0B08,0x71
0x52,0x0C08,0x71
0x52,0x0D08,0x71
0x52,0x0E08,0x71
# A 30Hz sync signal is then expected at Ser (e.g., MAX96705) GPO pin. No
sensor/PCLK is required.
```

44.10.7.1.1.2 Internal FSYNC (GMSL2)

This example demonstrates the programming sequence required to enable internal FSYNC in GMSL2 mode.

```
# Enable Internal FSYNC manual mode
0x52,0x04A0,0x04
# Turn off auto master link selection
0x52,0x04A2,0x00
# Disable overlap window (Default in rev D)
0x52,0x04AA,0x00
0x52,0x04AB,0x00
# Set to use 25MHz XTAL; AUTO_FS_LINKS = 0, FS_USE_XTAL = 1, FS_LINK_x[3:0] = 0,
GMSL2
0x52,0x04AF,0xC0
# Set FSYNC period to 25M/30 clock cycles. CLK = 25MHz. Sync freq = 30Hz
0x52,0x04A7,0x0C
0x52,0x04A6,0xB7
0x52,0x04A5,0x35
# Set FSYNC TX ID to 1 to match MFP1 on Ser side.
0x52,0x04B1,0x08
# Enable GPIO_RX_EN on Ser MFP1
0x80,0x02C1,0x84
# A 30Hz sync signal is then expected at Ser (e.g., MAX9295A) MFP1. No sensor/PCLK
is required.
```

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Distribution only under NDA
Provided to GMSL2 Customers Under NDA

44.10.7.1.1.3 External FSYNC (GMSL1)

The follow programming example enables external FSYNC in GMSL1 mode.

```
# This is GMSL1 Ext. FSYNC example; GPI_1/GPO_1 are used.
# Set Internal FSYNC off, GPIO is used for FSYNC, type = GMSL1
0x52,0x04A0,0x08
0x52,0x04AF,0x1F
# Enable GPI/GPO function in GMSL1 mode for each link; Select GPI_1 (MFP2) for
MAX96712/722
0x52,0x0B08,0x61
0x52,0x0C08,0x61
0x52,0x0D08,0x61
0x52,0x0E08,0x61
```

44.10.7.1.1.4 External FSYNC (GMSL2)

The following script configures and enables external FSYNC in GMSL2 mode.

```
# This is GMSL2 Ext. FSYNC example; MAX96712/722 MFP2/SER MFP1 are used for
GPI/GPO.
# Update SER MFP1 RX ID = 2 to match MFP2 of MAX96712/722
0x80,0x02C3,0x02
# Config SER MFP1 to forward GPIO from the MAX96712/722
0x80,0x02C1,0x84
# Set Internal FSYNC off, GPIO is used for FSYNC, type = GMSL2
0x52,0x04A0,0x08
0x52,0x04AF,0x9F
# Config MAX96712/722 MFP2 to receive external FSYNC signal for each link
0x52,0x0306,0x83
0x52,0x033D,0x22
0x52,0x0374,0x22
0x52,0x03AA,0x22
```

ADI Confidential for New vision under NDA
Analog Devices Confidential Do not to anyone under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.8 Video Pattern Generator (VPG)

The GMSL2 CSI-2 quad deserializers have two internal video pattern generators (VPG) that accommodate a wide range of resolutions and frame rates. The VPGs do not require an external PCLK source from the CSI input, and use the external 25MHz crystal clock (i.e., REFCLK input) to derive four different pixel clock (PCLK) options. Link lock is not required for the VPG to be used.

Each VPG has its own register block settings for timing configurations. VPG0 is connected after video pipes 0 to 3; VPG1 is connected after video pipes 4 to 7. Table 245 contains video pattern register addresses for both VPG0 and VPG1.

There are two clock configuration registers that set the PCLK value for the video pipes. The video pattern PCLK frequency can optionally be set 25/75MHz for all eight pipes or 150/375MHz on a per pipe basis. See Table 246 for configuration details. Table 247 contains reference information for VPG PCLK selection.

For additional details, refer to Video Timing Generator (VTG) and Video Pattern Generator (VPG) section. The GMSL SerDes GUI can be used to setup the VPG and to generate VPG register write example codes.

Table 245. Video Pattern Registers

Register (VPG0/VPG1)	Bits	Default Value	Description
0x1050/1080	7:0	0x03	**Pattern Generator 0/1 Register:** Bit 7: Generate VS according to the timing setting. Bit 6: Generate HS according to the timing setting. Bit 5: Generate DE according to the timing setting. Bit 4: Invert VSYNC of Video Timing Generator. Bit 3: Invert HSYNC of Video Timing Generator. Bit 2: Invert DE of Video Timing Generator. Bits [1:0]: Video Interface Timing Generation Mode.
0x1051/1081	5:4	00	**Pattern Generator Mode Register:** Bits [5:4]: Pattern selection. 01 = Color Gradient 10 = Checkerboard
0x1052/1082	7:0	0x00	**VS Delay Register 2:** VS Delay in terms of PCLK cycles. (Bits [23:16])
0x1053/1083	7:0	0x00	**VS Delay Register 1:** VS Delay in terms of PCLK cycles. (Bits [15:8])
0x1054/1084	7:0	0x00	**VS Delay Register 0:** VS Delay in terms of PCLK cycles. (Bits [7:0])
0x1055/1085	7:0	0x00	**VS High Register 2:** VS High Period in terms of PCLK cycles. (Bits [23:16])
0x1056/1086	7:0	0x00	**VS High Register 1:** VS High Period in terms of PCLK cycles. (Bits [15:8])
0x1057/1087	7:0	0x00	**VS High Register 0:** VS High Period in terms of PCLK cycles. (Bits [7:0])
0x1058/1088	7:0	0x00	**VS Low Register 2:** VS Low Period in terms of PCLK cycles. (Bits [23:16])
0x1059/1089	7:0	0x00	**VS Low Register 1:** VS Low Period in terms of PCLK cycles. (Bits [15:8])
0x105A/108A	7:0	0x00	**VS Low Register 0:** VS Low Period in terms of PCLK cycles. (Bits [7:0])

ADI Confidential for New Vision under NDA
Maxim Integrated Confidential, Provided to GMSL2_Customers_Only_NDA

0x105B/108B	7:0	0x00	**V2H Register 2:** VS edge to the rising edge of the first HS in terms of PCLK cycles. (Bits [23:16])
0x105C/108C	7:0	0x00	**V2H Register 1:** VS edge to the rising edge of the first HS in terms of PCLK cycles. (Bits [15:8])
0x105D/108D	7:0	0x00	**V2H Register 0:** VS edge to the rising edge of the first HS in terms of PCLK cycles. (Bits [7:0])
0x105E/108E	7:0	0x00	**HS High Register 1:** HS High Period in terms of PCLK cycles. (Bits [15:8])
0x105F/108F	7:0	0x00	**HS High Register 0:** HS High Period in terms of PCLK cycles. (Bits [7:0])
0x1060/1090	7:0	0x00	**HS Low Register 1:** HS Low Period in terms of PCLK cycles. (Bits [15:8])
0x1061/1091	7:0	0x00	**HS Low Register 0:** HS Low Period in terms of PCLK cycles. (Bits [7:0])
0x1062/1092	7:0	0x00	**HS Count Register 1:** HS pulses per frame. (Bits [15:8])
0x1063/1093	7:0	0x00	**HS Count Register 0:** HS pulses per frame. (Bits [7:0])
0x1064/1094	7:0	0x00	**V2D Register 2:** VS edge to the rising edge of the first DE in terms of PCLK cycles. (Bits [23:16])
0x1065/1095	7:0	0x00	**V2D Register 1:** VS edge to the rising edge of the first DE in terms of PCLK cycles. (Bits [15:8])
0x1066/1096	7:0	0x00	**V2D Register 0:** VS edge to the rising edge of the first DE in terms of PCLK cycles. (Bits [7:0])
0x1067/1097	7:0	0x00	**DE High Register 1:** DE High Period in terms of PCLK cycles. (Bits [15:8])
0x1068/1098	7:0	0x00	**DE High Register 0:** DE High Period in terms of PCLK cycles. (Bits [7:0])
0x1069/1099	7:0	0x00	**DE Low Register 1:** DE Low Period in terms of PCLK cycles. (Bits [15:8])
0x106A/109A	7:0	0x00	**DE Low Register 0:** DE Low Period in terms of PCLK cycles. (Bits [7:0])
0x106B/109B	7:0	0x00	**DE Count Register 1:** DE pulses per frame. (Bits [15:8])
0x106C/109C	7:0	0x00	**DE Count Register 0:** DE pulses per frame. (Bits [7:0])
0x106D/109D	7:0	0x00	**Gradient Increment Register:** Set color gradient increment.
0x106E/109E	7:0	0x00	**CHRK_COLOR_A_L Register:** Checkerboard mode color A low byte. RGB888 color Red (0-255).
0x106F/109F	7:0	0x00	**CHRK_COLOR_A_M Register:** Checkerboard mode color A mid byte. RGB888 color Green (0-255).
0x1070/10A0	7:0	0x00	**CHRK_COLOR_A_H Register:**

ADI Confidential for NDA Version under
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

			Checkerboard mode color A high byte. RGB888 color Blue (0-255).
0x1071/10A1	7:0	0x00	**CHRK_COLOR_B_L Register:** Checkerboard mode color B low byte. RGB888 color Red (0-255).
0x1072/10A2	7:0	0x00	**CHRK_COLOR_B_M Register:** Checkerboard mode color B mid byte. RGB888 color Green (0-255).
0x1073/10A3	7:0	0x00	**CHRK_COLOR_B_H Register:** Checkerboard mode color B high byte. RGB888 color Blue (0-255).
0x1074/10A4	7:0	0x00	**CHRK_RPT_A Register:** Checkerboard mode color A repeat count.
0x1075/10A5	7:0	0x00	**CHRK_RPT_B Register:** Checkerboard mode color B repeat count.
0x1076/10A6	7:0	0x00	**CHRK_ALT Register:** Checkerboard mode alternate line count.

Table 246. PLCK Settings Registers

Register	Bits	Default Value	Description
0x0009	1:0	00	**PCLK Freq Register:** Bit [1:0]: Set video pattern PCLK frequency. 00 = 25MHz 01 = 75MHz 10 = 150MHz (PATGEN_CLK_SRC = 0) 11 = 375MHz (PATGEN_CLK_SRC = 1)
0x01DC	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 0. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.
0x01FC	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 1. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.
0x021C	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 2. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.
0x023C	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 3. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.
0x025C	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 4. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.
0x027C	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 5. 0 = 150MHz, 1 = 375MHz

ADI Confidential for Review under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customer Under NDA

			Work together with Pattern CLK Freq register.
0x029C	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 6. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.
0x02BC	7	1	**PATGEN_CLK_SRC Register:** Select clock source for video pattern on pipe 7. 0 = 150MHz, 1 = 375MHz Work together with Pattern CLK Freq register.

Below is a reference table for VPG PCLK selection (Table 247). This is an alternative method of setting the pattern generator clock frequency with additional options for the PCLK frequency.

Table 247. Video Pattern PCLK Selection

Bitfield Name (Reg #)		PCLK Frequency
DEBUG_EXTRA (0x09 [1:0])	**PATGEN_CLK_SRC (0x1DC [7])**	
00	X	25MHz
01	X	75MHz
1x	0	150MHz
1x	1 (Default)	375MHz

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distributed under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.10.9 MIPI PRBS Generator

MIPI PRBS generators operating at 80MHz-2.5GHz are available in both C-PHY and D-PHY modes for eye diagram and benchmark testing. The generators have PRBS-9, -11, and -18 modes that are individually implemented on each of the eight MIPI lanes. The default seed value is based on the MIPI C-PHY v1.0 specification; a custom seed value can be optionally programmed and enabled. See Table 248 for relevant register information. In order to output the selected pattern, the force_csi_out_en bit must be set in register 0x08A0.

Table 248. MIPI PRBS Generator Registers

Register	Bits	Default Value	Description	Decode
0x08C0	1:0	0x0	PRBS enable for PHY0 lane 0 eye diagram.	**MIPI_PRBS_EN_P*_LN*** 00: Disabled 01: PRBS 9 enabled 10: PRBS 11 enabled 11: PRBS 18 enable
	3:2	0x0	PRBS enable for PHY0 lane 1 eye diagram.	
	5:4	0x0	PRBS enable for PHY1 lane 0 eye diagram.	
	7:6	0x0	PRBS enable for PHY1 lane 1 eye diagram.	
0x08C1	1:0	0x0	PRBS enable for PHY2 lane 0 eye diagram.	
	3:2	0x0	PRBS enable for PHY2 lane 1 eye diagram.	
	5:4	0x0	PRBS enable for PHY3 lane 0 eye diagram.	
	7:6	0x0	PRBS enable for PHY3 lane 1 eye diagram.	
0x08C2	0	0x0	Custom seed enable for PHY0 lane 0.	**MIPI_CUST_SEED_EN_P*_LN*** 0: Disabled 1: Enabled
	1	0x0	Custom seed enable for PHY0 lane 1.	
	2	0x0	Custom seed enable for PHY1 lane 0.	
	3	0x0	Custom seed enable for PHY1 lane 1.	
	4	0x0	Custom seed enable for PHY2 lane 0.	
	5	0x0	Custom seed enable for PHY2 lane 1.	
	6	0x0	Custom seed enable for PHY3 lane 0.	
	7	0x0	Custom seed enable for PHY3 lane 1.	
0x08C3	1:0	0x2	Custom seed [17:16] for MIPI PRBS eye diagrams. Default reset is from C-PHY specification.	**MIPI_CUSTOM_SEED_2** Bits [17:16]: Reset value for PRBS if enabled by MIPI custom seed enable register.
0x08C4	7:0	0x78	Custom seed [15:8] for MIPI PRBS eye diagrams. Default reset is from C-PHY specification.	**MIPI_CUSTOM_SEED_1** Bits [15:8]: Reset value for PRBS if enabled by MIPI custom seed enable register.
0x08C5	7:0	0x9a	Custom seed [7:0] for MIPI PRBS eye diagrams. Default reset is from C-PHY specification.	**MIPI_CUSTOM_SEED_0** Bits [7:0]: Reset value for PRBS if enabled by MIPI custom seed enable register.
0x08A0	7	0	Force MIPI clock output, even if no video present.	0: normal operation 1: Force enable all MIPI clocks

ADI Confidential for Review Only
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

44.10.10 GMSL2 Video PRBS Generator and Checker

Serial link systems incorporating a GMSL2 CSI-2 quad deserializer and GMSL2 serializer have several options for generating and checking video PRBS packets to verify that a significant portion of the data path between a serializer and the quad deserializer is error-free. First, the GMSL2 CSI-2 quad deserializer can both generate and check CSI-2 packetized 24bpp PRBS video packets. The quad deserializer generates valid CSI-2 video packets in PRBS-7, -9, or -24 format that are sent to the checker in the quad deserializer via a loopback path through the connected MIPI serializer.

Note: The quad deserializer features 25, 75, 150, and 375MHz clock sources.

The method of conducting VPRBS testing is dependent on the system and serializer connected. For example, not all serializers can create the required PCLK signal to transmit a VPRBS signal. Serializers like the MAX96717 can create an internal PCLK signal, where MAX9295A serializers must receive an external PCLK signal from the system. A common system variation that causes a difference in the VPRBS testing method is whether image sensor data is active and transmitting though the serializer or not. An example of VPRBS testing for serializers that internally generate PCLK are included in this document as well as an example of conducting VPRBS testing as video is running.

It is not recommended to conduct video PRBS testing in some quad deserializer systems as external connections may be required to obtain the required serializer PCLK. The test can be useful for bringups and testing where the external connections can be made to the serializer. The MIPI input to the serializer must also be disabled prior to running this test. See the *PRBS Testing* section for additional information.

Each of the eight quad deserializer video pipes have independent control over packet generation and checking parameters and dedicated error count registers. Reference Table 249 for register information.

Table 249. MIPI Video PRBS Generator and Checker Registers

Register	Bits	Default Value	Description	Decode
0x01DC **0x01FC** **0x021C** **0x023C** **0x025C** **0x027C** **0x029C** **0x02BC**	7	0x1	Pattern generator clock source for video PRBS7, PRBS9, PRBS24, checkerboard, and gradient patterns.	0b0: 150MHz 0b1: 375MHz (default)
	4	0b0	Enables video PRBS24 generator/checker	0b0: Disabled 0b1: Enabled
	3	0b0	Enables video PRBS7 generator/checker	0b0: Disabled 0b1: Enabled
	2	0b0	Enables video PRBS9 generator/checker	0b0: Disabled 0b1: Enabled
0x01DB **0x01FB** **0x021B** **0x023B** **0x025B** **0x027B** **0x029B** **0x02BB**	7:0	0x00	Video PRBS error counter, clears on read	0xXX: Number of video PRBS errors since last read

ADI Confidential for New Version under NDA
Maxim Integrated Company Confidential, Provided to GMSL2 Customers Under NDA

0x0029	2	0b1	Enables reporting of video PRBS errors (VPRBS_ERR_FLAG—0x2A) at ERRB pin.	0b0: Disable video PRBS error reporting 0b1: Enable video PRBS error reporting
0x002A	2	0b0	Video PRBS Error Flag. Asserted when VPRBS_ERR (0x1DB) > 0.	0b0: VPRBS_ERR ≤ 0 0b1: VPRBS_ERR > 0

Note: MIPI Video PRBS errors can be reported to the ERRB pin; reference the GMSL2 Error Reporting (ERRB Pin) section for additional information.

44.10.10.1.1 Programming Examples

44.10.10.1.1.1 Loopback VPRBS

The following script will configure a single MAX96717 and a MAX96712 EVkit to be in a loopback configuration for PHY 1 in 2x4 mode. This configuration may require removal of I2C connections from the highspeed connector of the MAX96717 EVK. This may also be achieved by disabling the MAX96717 local I2C through programming before connecting in loopback configuration.

```
# Connect a MAX96717 and a MAX96712 EVKit together through the highspeed connector.
# Connect Serializer GMSL link to Deserializer GMSL link B

#Enable Deserializer link B only
0x52,0x0006,0xF2
#Correct Deserializer lane mapping to be 0xE4
0x52,0x08A3,0xE4
#Reduce deserializer PHY1 output MIPI rate to 500Mbps per lane
0x52,0x0418,0x25
#Enable force_CSI_OUT for 2x4 mode
0x52,0x08A0,0x84
#Enable PRBS(2^24) generator and checker for deserializer pipe 1
0x52,0x01FC,0x90

#Verify serializer has PCLKDET = 1
0x80,0x0112,0x8A
#Verify deserializer has VIDEO_LOCK = 1 for Pipe 1
0x52,0x01FC,0x91
#Verify deserializer does not have PRBS errors VPBRB_ERR = 0x00
0x52,0x01FB,0x00

#Optional Step: Enable Serializer errors to verify setup and PRBS error detection
#Note: Error generation also creates decode and idle errors, so these must be
cleared as well.
#Enable serializer error generator
0x80,0x0029,0x18
#Delay for short time to accumulate errors.
#Disable serializer error generator
0x80,0x0029,0x08
#Verify deserializer has PRBS errors VPBRB_ERR > 0x00, 0xFF is used as the read
value in this example, but errors may vary. VPRBS Errors should clear after reading
this register.
```

ADI Confidential for New Vision under NDA

```
0x52,0x01FB,0xFF
```

44.10.10.1.1.2 Serializer Generated VPRBS

The following script will configure a single MAX96717 and a MAX96712 EVkit to conduct the standard VPRBS test. In this test the serializer will generate a PRBS pattern and the deserializer will check it. This test requires that the serializer has PCLK and that no video is running into the serializer.

```
# Connect Serializer GMSL link to Deserializer GMSL link B

#Enable Deserializer link B only
0x52,0x0006,0xF2
#Disable auto bpp on serializer
0x80,0x0110,0x60
#Enable serializer internal PCLK generation, PCLK = 150MHz
0x80,0x024F,0x0D
#Enable serializer pattern generator
0x80,0x026B,0x01
#Delay for 3ms
#Enable PRBS(2^24) checker for deserializer pipe 1
0x52,0x01FC,0x10
#Delay for 3ms
#Enable serializer VPRBS generator
0x80,0x026B,0x81

#Verify serializer has PCLKDET = 1
0x80,0x0112,0x8A
#Verify deserializer has VIDEO_LOCK = 1 for Pipe 1
0x52,0x01FC,0x11
#Verify deserializer does not have PRBS errors VPBRB_ERR = 0x00
0x52,0x01FB,0x00

#Optional Step: Enable Serializer errors to verify setup and PRBS error detection
#Note: Error generation also creates decode and idle errors, so these must be
cleared as well.
#Enable serializer error generator
0x80,0x0029,0x18
#Delay for short time to accumulate errors.
#Disable serializer error generator
0x80,0x0029,0x08
#Verify deserializer has PRBS errors VPBRB_ERR > 0x00, 0xFF is used as the read
value in this example, but errors may vary. VPRBS Errors should clear after reading
this register.
0x52,0x01FB,0xFF
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL Customers Under NDA

44.10.10.1.1.3 Serializer Generated VPRBS (Video Running)

The following script will configure a single MAX9295A and a MAX96712 EVkit to conduct the standard VPRBS test while video is running. In this test the serializer will generate a PRBS pattern and the deserializer will check it. This test requires a special sequence to avoid image sensor data corrupting the VPRBS signal.

```
# Connect Serializer GMSL link to Deserializer GMSL link, program for normal
# operation, and start image sensor. This Example uses Link B only.
# Ensure that serializer detects PCLK and Deserializer has video lock.
# Configure the SerDes for VPRBS testing.

#Enable Deserializer link B only
0x52,0x0006,0xF2
#Verify serializer has PCLKDET = 1 on pip X
0x80,0x0102,0x8A
#Verify deserializer has VIDEO_LOCK = 1 for Pipe 1
0x52,0x01FC,0x11
#Disable serializer transmit VID_TX_EN_X = 0
0x80,0x0002,0x43
#IF: serializer AUTO_BPP != 1, set soft_bpp = 24 and soft_bpp_en = 1
#This step is only required if BPP is not 24bpp.
0x80,0x031C,0x38
#Setup Pattern Generator on serializer for pipe X.
0x80,0x01CF,0x10
0x80,0x01D0,0x00
0x80,0x01D1,0xFF
0x80,0x01D2,0xFF
0x80,0x01DC,0x00
0x80,0x01DD,0x00
0x80,0x01DE,0xFF
0x80,0x01DF,0x00
0x80,0x01E0,0x10
0x80,0x01E1,0x00
0x80,0x01E2,0xFF
0x80,0x01E3,0x00
0x80,0x01E4,0x03
0x80,0x01E7,0xFF
0x80,0x01E8,0xFF
0x80,0x01E9,0xFF
0x80,0x01C8,0xA3
0x80,0x01E5,0x01
Enable serializer transmit VID_TX_EN_X = 1
0x80,0x0002,0x13
#Verify serializer has PCLKDET = 1
0x80,0x0102,0x8A
#Verify deserializer has VIDEO_LOCK = 1 for Pipe 1
0x52,0x01FC,0x01
#Enable PRBS(2^24) checker for deserializer pipe 1
0x52,0x01FC,0x11
#Enable serializer VPRBS generator
0x80,0x01E5,0x81
#Verify Deserializer has LOCKED_B= 1
0x52,0x000A,0x80
#Verify deserializer has VIDEO_LOCK = 1 for Pipe 1
0x52,0x01FC,0x11
#Read deserializer PRBS errors to clear them VPRBS _ERR = 0xFF
0x52,0x01FB,0xFF
```

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Company Confidential, Provided to GMSL2 Customers Under NDA

```
#Read deserializer PRBS errors VPRBS _ERR = 0x00
#There should be no errors at this point in reading the errors.
#IF: VPRBS _ERR = 0, continue with VPRBS test
#ELSE: VPRBS _ERR > 0 go to restart VPRBS generator section:
      There can be two reasons for getting VPRBS_ERR > 0.
      1) There might be a synchronization issue
      2) The GMSL2 link might not be error free
            To distinguish this, users should read error registers (decoding, idle,
            lcrc, vid_pxl_crc)
            1) If these errors are all zero, it means checker is not synchronized
            with generator. Go to restart VPRBS generator section
            2) If these errors are non-zero, it means checker is synchronized with
            generator but the GMLS2 system is not error free.

# VPRBS _ERR = 0, errors cleared successfully
0x52,0x01FB,0x00
#Setup error generator
0x80,0x002A,0xA0
#Enable serializer error generator
0x80,0x0029,0x10
#Delay for 3ms
#Disable serializer error generator
0x80,0x0029,0x00
# VPRBS_ERR should be > 0. If VPRBS = 0 an error occurred. Go to restart VPRBS
generator section.
0x52,0x01FB,0x00
# After clearing device errors: VPRBS errors, decode and idle errors, pixel and
line CRC errors to clear them, users may allow the test to run for an amount of
time in order to check BER.

#Restart VPRBS generator section
#Disable serializer VPRBS generator
0x80,0x01E5,0x01
#Disable PRBS(2^24) checker for deserializer pipe 1
0x52,0x01FC,0x00
#Enable PRBS(2^24) checker for deserializer pipe 1
0x52,0x01FC,0x11
#Enable serializer VPRBS generator
0x80,0x01E5,0x81
#Verify deserializer has VIDEO_LOCK = 1 for Pipe 1
0x52,0x01FC,0x11
#Verify deserializer has LOCKED_B= 1
0x52,0x000A,0x80
#Read deserializer PRBS errors to clear them VPRBS _ERR = 0xFF
0x52,0x01FB,0xFF
# VPRBS _ERR = 0, errors cleared successfully
0x52,0x01FB,0x00
# User may now continue with the VPRBS test.
```

Maxim Integrated Confidential, Provided to GMSL2_Customers Under NDA
Analog Devices Confidential Distribution only under NDA
ADI Confidential for Newvision under NDA

44.10.11 I2C/UART Control Channels

Most GMSL2 devices provide a single I2C/UART control channel and a single I2C/UART control channel port. However, MAX96712 provides three independent I2C/UART control channels and control channel ports (Port 0, Port 1, and Port 2). Device registers can be accessed locally from any of the three CC ports. The primary I2C port can access the deserializer, connected serializers, and the remote peripherals connected to the serializer's control channel port. The secondary ports have access to the deserializer and remote-side devices but cannot access serializer registers. All ports may be utilized at the same time, if only one port is accessing deserializer registers.

Arbitration between the ports is performed on a first come first served basis; lower indexed ports are given highest priority in the case of a simultaneous access. Due to this parallel communication is not supported for pass-through ports. In I2C mode, the deserializer will pend I2C accesses with clock stretching until the current I2C host access is complete and an I2C stop condition is received. The operation is the same for UART mode, except that the deserializer waits 48-bits time following the stop bit of the last UART frame before resuming UART accesses. To disconnect local port register access, the port must be disabled. In UART mode, local register access can be disabled by forcing the port into bypass mode.

Note: Each I2C port can be simultaneously routed to any combination of GMSL links; however, each UART port can be routed to only one GMSL link at a time.

By default, Port 0 is the primary, full-featured I2C/UART port with remote device programming access across the GMSL2 link. Ports 1 and 2 (i.e., the secondary I2C/UART ports) can be tunneled to the remote side of the link and used as pass-through channels for peripheral communication. However, the control channel crossover feature (programmed with CC_CROSSOVER_SEL) allows Port 1 or 2 to be configured as the primary (main) control channel and enables crossover access to the remote serializer registers.

The three I2C/UART ports have independently configured access to the control channel and pass-through channels for each of the four GMSL links (Figure 223). Each link's pass-through channels can be separately enabled or disabled, providing extensive routing flexibility between the I2C/UART ports and the remote device(s). This is especially useful in UART applications. Ports operating in I2C mode can communicate over any number of available GMSL2 links, whereas ports operating in UART mode can only communicate over one link at a time. Since UART communications cannot be broadcasted across multiple links, remote devices must be programmed over one link at a time. The UART link selection registers (UART_PORT{0-2}_LINK_SELECT) are used to enable UART communication on the selected link.

ADI Confidential for New Vision under NDA
Analog Devices Confidential to Maxim Integrated Partners Under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

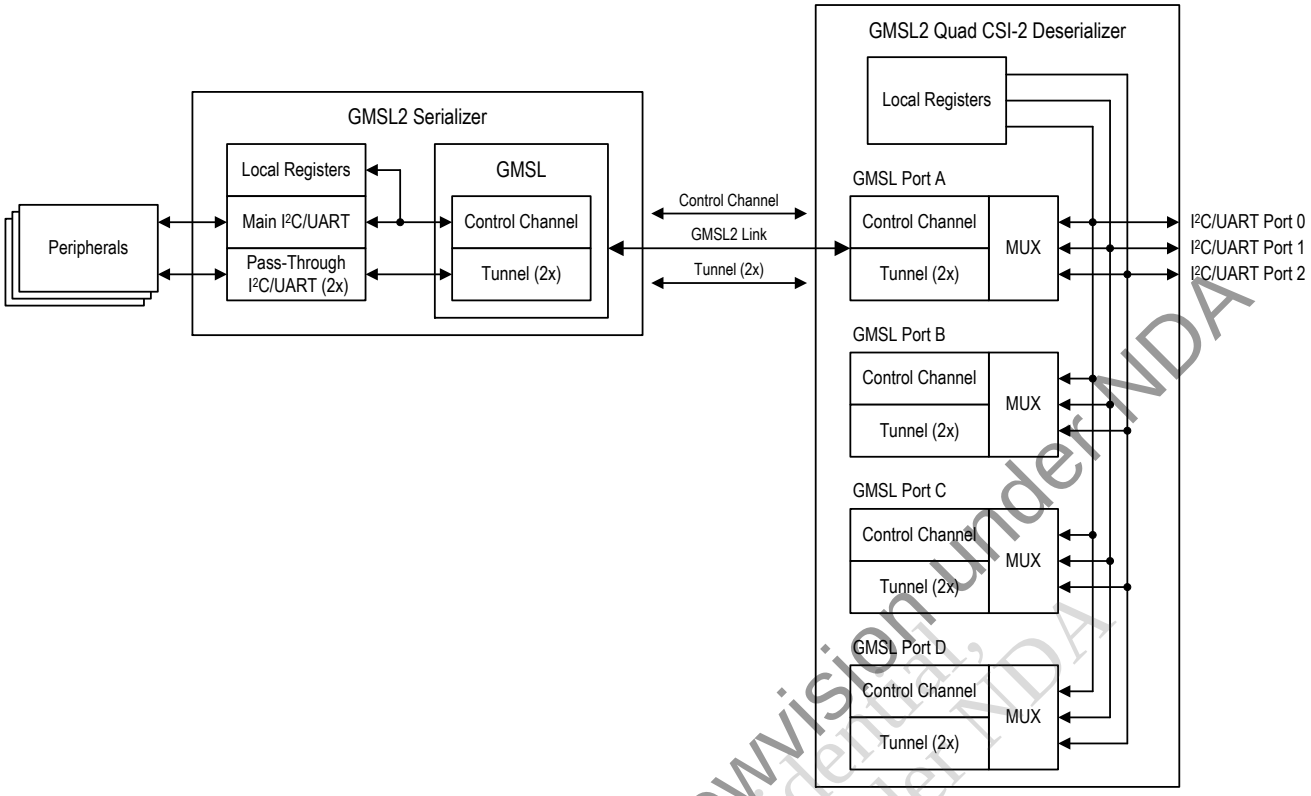


Figure 223. CSI-2 Quad Deserializer I2C/UART Routing

ADI Confidential for Newvision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.10.11.1 I2C/UART Splitter Mode and CSI-2 Quad Deserializers

Four serializer devices can connect to the CSI-2 quad deserializer via four independent serial links. Remote downstream device access can be modified with the DIS_REM_CC[0] register, but only crossover selection can change the port access to the serializer.

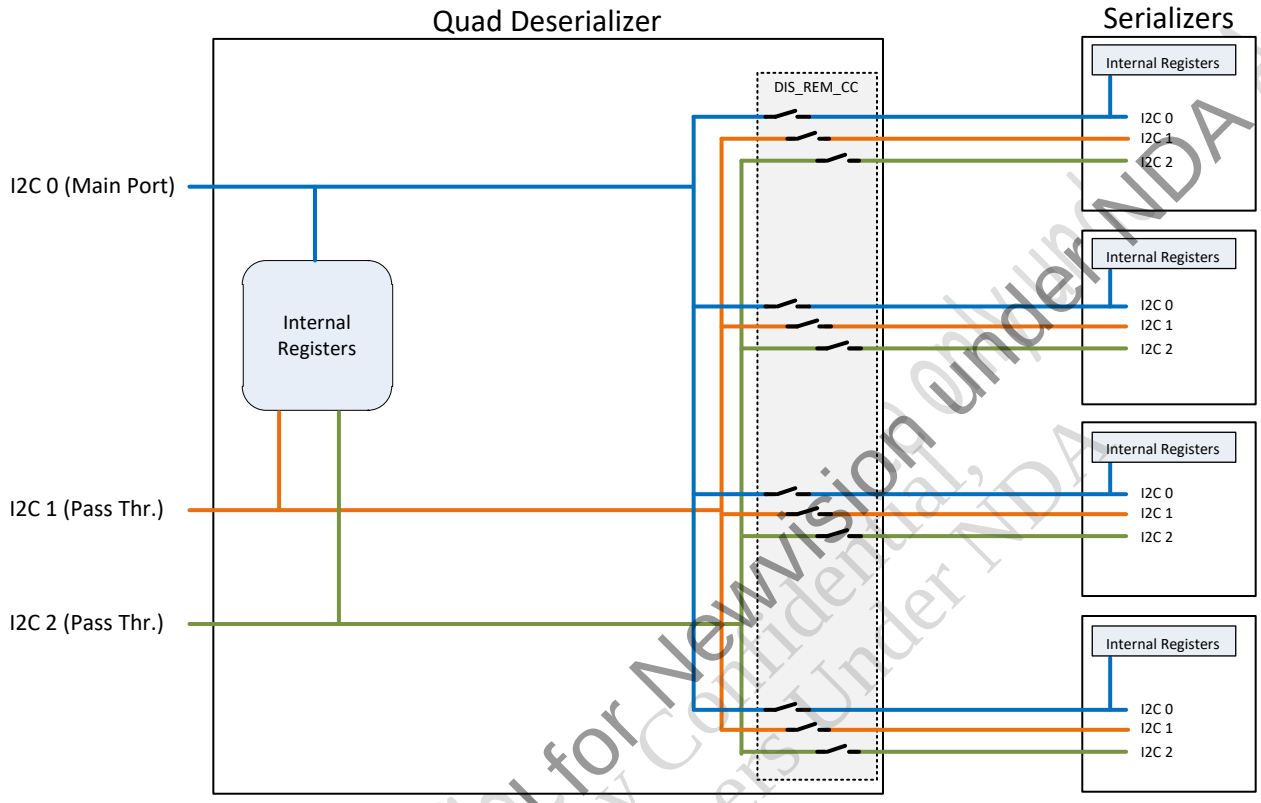


Figure 224. Quad Deserializer I2C Port access

Note: The pass-through channels of connected GMSL2 serializers have access to internal CSI-2 quad deserializer registers.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Only
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.11.2 I2C/UART Control Channel Crossover

GMSL2 CSI-2 quad deserializers features a control channel crossover switch that allows the selection of a new primary port, both for redundancy and multi-master operation. It can be enabled in either GMSL1 or GMSL2 mode.

By default, Port 0 is the primary, full-featured I2C/UART port with remote device programming access across the serial link. Ports 1 and 2 (i.e., the secondary I2C/UART ports) can be tunneled to the remote side of the link and used as pass-through channels for peripheral communication (see *Pass-Through I2C/UART*). Reference Figure 223 for control channel routing.

- Port 0: I2C_0 (MFP15/16)
- Port 1: I2C_1 (MFP11/12)
- Port 2: I2C_2 (MFP04/14)

Note: All three I2C/UART ports have access to the internal deserializer registers and can change the device operation. Use care to limit deserializer overwrites. It is the responsibility of the system designer to prevent a single I2C host from monopolizing I2C accesses with repeated start conditions. I2C accesses must be balanced using timely stop conditions by each host controller.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential, Under NDA
Provided to GMSL2_Customers Under NDA

The control channel crossover feature allows port 1 or 2 to be configured as the main control channel port and enables crossover access to the primary serializer I2C channel and serializer registers. When the crossover is active, the selected crossover port is switched internally with primary Port 0. See *Figure 225* for an example of crossover switching. Relevant control channel registers are presented in *Table 250*.

Note: Only one primary I2C control channel can be active at a time.

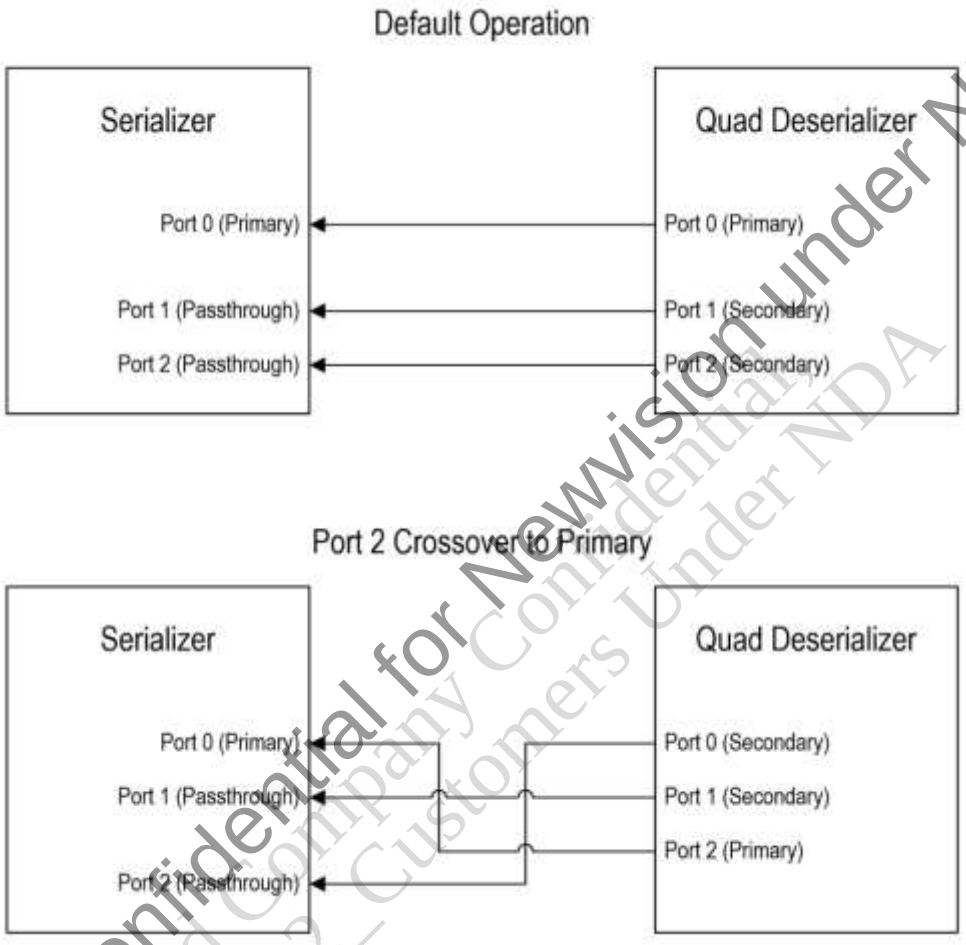


Figure 225. CSI-2 Quad Deserializer I2C/UART Crossover Switching

ADI Confidential for Newvision under NDA
Analog Devices Confidential NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

Table 250. Control Channel Registers

Register	Bits	Default Value	Description
0x0001	7:0	0x04	**I2C/UART CC Register:** Bits [7:6]: IIC_SEL; Bits are set according to the latched CFG pin value at power up. Bit 7 for port 1 Bit 6 for port 0 0 = UART, 1 = I2C Bits [5:4]: Disable control channel for port 0 and 1. Bit 5 for port 1 Bit 4 for port 0 0 = Connected, 1 = Disabled Bit 3: IIC_SEL_P2; Select I2C/UART for port 2 0 = UART, 1 = I2C Bit 2: Disable control channel for port 2. 0 = Connected, 1 = Disabled (by default) Bit 2: ALT_II2C_SEL 0 = Select MFP11 and 12 for I2C/UART port 1. 1 = Select MFP8 and 9 for I2C/UART port 1. Bit 1: FAILOVER 0 = Normal mode 1 = Port 0 failover to port 1
0x0003	7:0	0xAA	**Disable Remote Control Channel Register:** Bits [7:6]: Remote control channel for link D. Bits [5:4]: Remote control channel for link C. Bits [3:2]: Remote control channel for link B. Bits [1:0]: Remote control channel for link A. x0 = Enable remote control channel from port 0. x1 = Disable remote control channel from port 0. 0x = Enable remote control channel from port 1. 1x = Disable remote control channel from port 1.
0x0007	7:0	0x00	**I2C Crossover Register:** Bit [3], Bit [7]: Crossover for link D Bit [2], Bit [6]: Crossover for link C Bit [1], Bit [5]: Crossover for link B Bit [0], Bit [4]: Crossover for link A 00 = No crossover (port 0 by default) 01 = Enable crossover for port 1 1x = Enable crossover for port 2
0x000E	3:0	1111	**Disable Remote Control Channel Register:** Bits 4: Remote control channel for link D. Bits 3: Remote control channel for link C. Bits 2: Remote control channel for link B. Bits 1: Remote control channel for link A. 0 = Enable remote control channel from port 2 1 = Disable remote control channel from port 2
0x0B04	4:3	00	**GMSL1 Link A Register:** Bits [4:3]: CC_PORT_SEL Selects which I2C/UART port is connected as the main control channel in GMSL1 mode. 00: Port 0 (RX0_SDA0, TX0_SCL0) 01: Port 1 (RX1_SDA1, TX1_SCL1)

ADI Confidential for New Version under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2_Customer_User_NDA

			10: Port 2 (RX2_SDA2, TX2_SCL2)
0x0C04	4:3	00	**GMSL1 Link B Register:** Bits [4:3]: CC_PORT_SEL Selects which I2C/UART port is connected as the main control channel in GMSL1 mode. 00: Port 0 (RX0_SDA0, TX0_SCL0) 01: Port 1 (RX1_SDA1, TX1_SCL1) 10: Port 2 (RX2_SDA2, TX2_SCL2)
0x0D04	4:3	00	**GMSL1 Link C Register:** Bits [4:3]: CC_PORT_SEL Selects which I2C/UART port is connected as the main control channel in GMSL1 mode. 00: Port 0 (RX0_SDA0, TX0_SCL0) 01: Port 1 (RX1_SDA1, TX1_SCL1) 10: Port 2 (RX2_SDA2, TX2_SCL2)
0x0E04	4:3	00	**GMSL1 Link D Register:** Bits [4:3]: CC_PORT_SEL Selects which I2C/UART port is connected as the main control channel in GMSL1 mode. 00: Port 0 (RX0_SDA0, TX0_SCL0) 01: Port 1 (RX1_SDA1, TX1_SCL1) 10: Port 2 (RX2_SDA2, TX2_SCL2)

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.10.11.2.1 Programming Examples

44.10.11.2.1.1 Enable Port 2

```
# Enable I2C port 2 for local register access
0x52,0x0001,0xC8
```

44.10.11.2.1.2 Enable Port 1 Crossover on Link A in GMSL2 Mode

```
# Switch I2C/UART port 0 to have the local access only
# Set I2C/UART port 1 to be the main control channel for both local and remote
# Disable link A remote access for port 0.
0x52,0x0003,0xA9
# Enable I2C port 1 crossover on link A
0x52,0x0007,0x10
```

44.10.11.2.1.3 Enable Port 2 Crossover on Link A in GMSL2 Mode

```
# Switch I2C/UART port 0/1 to have the local access only
# Set I2C/UART port 2 to be the main control channel for both local and remote
# Enable I2C port 2 (rev D has this bit 3 IIC_SEL_P2 set by default)
0x52,0x0001,0xC8
# Disable link A remote for port 0/1
0x52,0x0003,0xAB
# Enable I2C port 2 crossover on link A
0x52,0x0007,0x01
0x52,0x000E,0x0E
```

44.10.11.2.1.4 Enable Port 1 Crossover on Link A in GMSL1 Mode

```
# Set CC_PORT_SEL = 01 for Port 1 crossover on Link A
0x52,0x0B04,0x0B
```

44.10.11.2.1.5 Enable Port 2 Crossover on Link C in GMSL1 Mode

```
# Set CC_PORT_SEL = 10 for Port 2 crossover on Link C
0x52,0x0D04,0x13
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute Under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.12 I2C Broadcasting

GMSL2 CSI-2 quad deserializers can broadcast to all addresses by default, but it is often the case that each device will require a unique address for individual programming and identification. Through I2C translation and address reassignment, each serializer and image sensor can have both a unique address and a broadcasting address. This allows for selective programming of each device and the ability to broadcast commands to all devices at the same time.

An example of I2C broadcasting is shown in the block diagram below (Figure 226). A GMSL2 CSI-2 quad deserializer is connected to four identical camera modules. Each camera module comprises a GMSL2 serializer at I2C address 0x80 and an image sensor at address 0x20. See *Table 251* and *Table 252* below for addressing information.

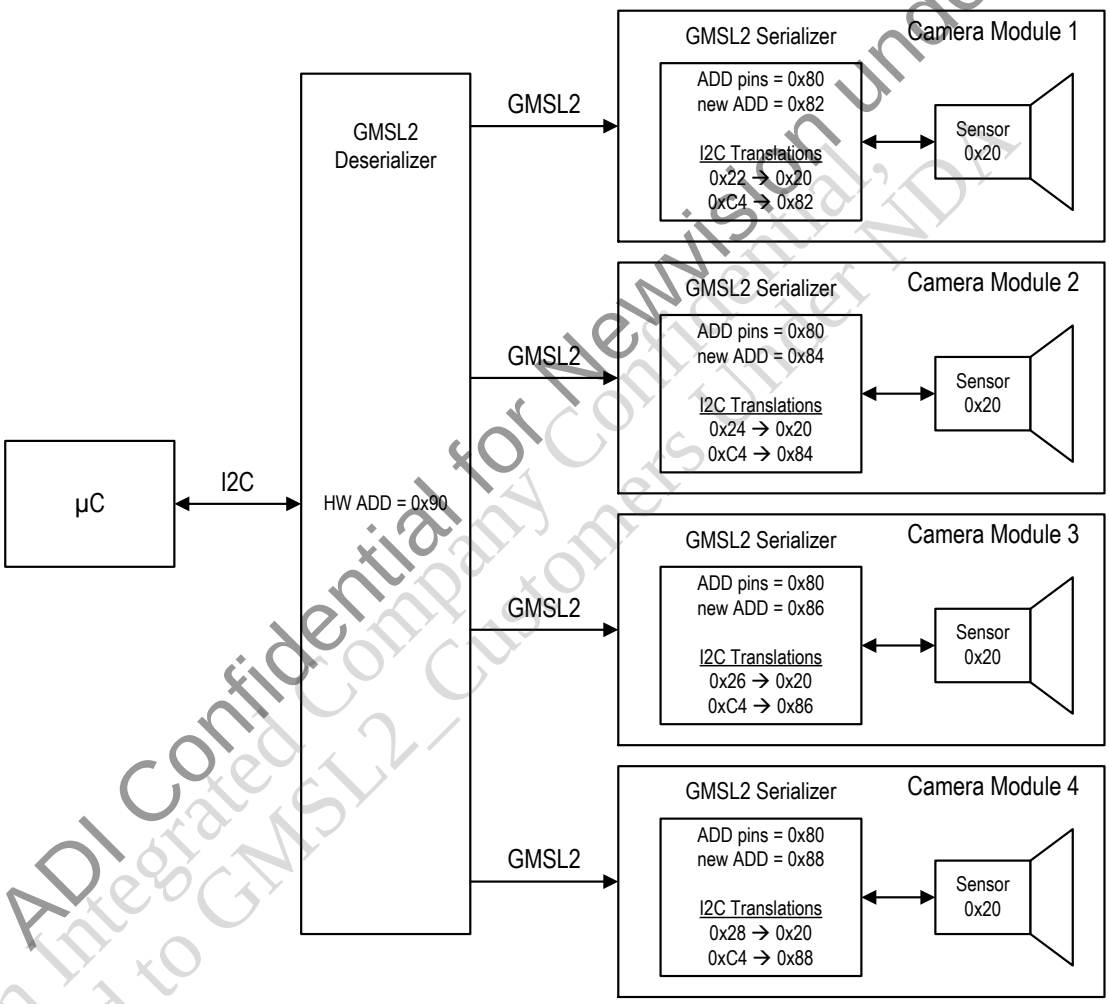


Figure 226. I2C Broadcasting (Quad Deserializer)

Analog Devices Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers_Under_NDA

The procedure for the I2C broadcasting example (Figure 226) is described below.

1) Isolate the camera module 1, by enabling link A only by setting LINK_EN_A[0] to 0b1.
2) Poll the LOCKED pin until it goes high.
3) Change the serializer device address in camera module 1 from 0x80 to 0x82. This is done with a register write to DEV_ADDR[6:0], located in REG0.
4) Modify the first address translation register in this serializer to give a broadcast address (0xC4) to the serializer. Program 0xC4 into the source register SRC_A[6:0], and 0x82 in the destination register DST_A[6:0]. Thus, for the serializer in camera module 1, anything sent to address 0xC4 will be sent to address 0x82 instead.
5) Modify the second translation register in this serializer to give a unique address to the image sensor. Program 0x22 into the source register SRC_B[6:0], and 0x20 into the destination register DST_B[6:0]. Thus, for the serializer in camera module 1, anything sent to address 0x22 will be sent to address 0x20 instead.
6) Isolate the camera module 2, by enabling link B only by setting LINK_EN_B[0] to 0b1.
7) Poll the LOCKED pin until it goes high.
8) Change the serializer device address in camera module 2 from 0x80 to 0x84. This is done with a register write to DEV_ADDR[6:0], located in REG0.
9) Modify the first address translation register in this serializer to give a broadcast address (0xC4) to the serializer. Program 0xC4 into the source register SRC_A[6:0], and 0x84 in the destination register DST_A[6:0]. Thus, for the serializer in camera module 2, anything sent to address 0xC4 will be sent to address 0x84 instead.
10) Modify the second translation register in this serializer to give a unique address to the image sensor. Program 0x24 into the source register SRC_B[6:0], and 0x20 into the destination register DST_B[6:0]. Thus, for the serializer in camera module 2, anything sent to address 0x24 will be sent to address 0x20 instead.
11) Isolate the camera module 3, by enabling link C only by setting LINK_EN_C[0] to 0b1.
12) Poll the LOCKED pin until it goes high.
13) Change the serializer device address in camera module 2 from 0x80 to 0x86. This is done with a register write to DEV_ADDR[6:0], located in REG0.
14) Modify the first address translation register in this serializer to give a broadcast address (0xC4) to the serializer. Program 0xC4 into the source register SRC_A[6:0], and 0x86 in the destination register DST_A[6:0]. Thus, for the serializer in camera module 3, anything sent to address 0xC4 will be sent to address 0x86 instead.
15) Modify the second translation register in this serializer to give a unique address to the image sensor. Program 0x26 into the source register SRC_B[6:0], and 0x20 into the destination register DST_B[6:0]. Thus, for the serializer in camera module 3, anything sent to address 0x26 will be sent to address 0x20 instead.
16) Isolate the camera module 4, by enabling link D only by setting LINK_EN_D[0] to 0b1.
17) Poll the LOCKED pin until it goes high.
18) Change the serializer device address in camera module 4 from 0x80 to 0x88. This is done with a register write to DEV_ADDR[6:0], located in REG0.
19) Modify the first address translation register in this serializer to give a broadcast address (0xC4) to the serializer. Program 0xC4 into the source register SRC_A[6:0], and 0x88 in the destination register DST_A[6:0]. Thus, for the serializer in camera module 4, anything sent to address 0xC4 will be sent to address 0x88 instead.
20) Modify the second translation register in this serializer to give a unique address to the image sensor. Program 0x28 into the source register SRC_B[6:0], and 0x20 into the destination register DST_B[6:0].

ADI Confidential for New Version under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

Thus, for the serializer in camera module 4, anything sent to address 0x28 will be sent to address 0x20 instead.

21) Now enable all the links by writing LINK_EN_A[0], LINK_EN_B[0], LINK_EN_B[0], and LINK_EN_D[0] to 0b1.
22) Perform link reset for each link by writing the self-clearing bit RESET_ONESHOT_A, RESET_ONESHOT_B, RESET_ONESHOT_C, and RESET_ONESHOT_D high.
23) Poll the LOCKED pin until it goes high.
24) All devices should be present on the I2C bus. Continue with any additional required system configuration.

The serializers are assigned a single device address to allow writes to all devices (Table 251).

Table 251. I2C Broadcasting (Quad) Example – Serializer

I2C Address	SRC_A	DST_A	Sink Device(s)
0x82	0xC4	0x82	Serializer in Camera Module 1
0x84	0xC4	0x84	Serializer in Camera Module 2
0x86	0xC4	0x86	Serializer in Camera Module 3
0x88	0xC4	0x88	Serializer in Camera Module 4

Each image sensor is assigned a unique device address (Table 252).

Table 252. I2C Broadcasting (Quad) Example – Image Sensor

I2C Address	SRC_B	DST_B	Sink Device(s)
0x20	0x22	0x20	Image Sensor in Camera Module 1
0x20	0x24	0x20	Image Sensor in Camera Module 2
0x20	0x26	0x20	Image Sensor in Camera Module 3
0x20	0x28	0x20	Image Sensor in Camera Module 4

ADI Confidential for Newvision Under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.10.13 Pass-Through I2C/UART

Pass-through I2C is used to create a direct I2C connection through the serial link while bypassing the serializer. For example, an I2C slave on the remote side (e.g., an EEPROM in a camera module) can be controlled by a different local master (e.g., a second SoC) directly. The I2C ports must be matched on both the serializer and deserializer sides for I2C pass-through to function.

44.10.13.1.1 Programming Example

This example shows how to setup pass-through I2C on ports 1 and 2 for the MAX96712 with the MAX96717 serializer

```
# Select I2C for all ports
0x52, 0x00, 0x01, 0xC8,
# Enable Remote channel for Port 1 for Link A I2C access
0x52, 0x00, 0x03, 0xA8,
# Enable Remote channel for Port 2 for Link A I2C access
0x52, 0x00, 0x0E, 0x0E,

# Enable Port 1 and Port 2 for pass-through I2C on serializer. This disables the
main I2C on serializer.
0x80, 0x00, 0x01, 0xE8,
*/
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.14 I2C Address Translation

GMSL2 CSI-2 quad deserializers are often connected to more than one serializer. By default, I2C communications are broadcasted if I2C-connected peripherals (i.e., serializers and cameras) share the same I2C address. Each link can be individually selected via register 0x06 to access each serializer or camera separately on the link (Table 253).

Note: I2C address translation is a feature of serializers; however, it is commonly used while working with the GMSL2 CSI-2 quad deserializers. I2C address translation permits access to individual links while retaining link connectivity and the physical I2C addresses of devices. Reference the I2C/UART section for serializer configuration information.

Both GMSL1 and GMSL2 serializers can support I2C address translation for up to two device addresses. In typical applications, one address translation is used for the serializer itself, while the other translation is used for its peripheral (e.g., a camera). I2C address translation requires specifying a source address (the address to translate from) and a destination address (the address to translate to). For example, a serializer can assign a new I2C address as the source for a camera while setting the camera's original address as the destination. All subsequent I2C commands to the camera on that particular link can then use the new I2C address.

Table 253. Link Initialization Registers Used for Link Selection

Register	Bits	Default Value	Description
0x0006	7:0	0xFF	**GMSL Link/PHY Enable and Mode Select Register:** Bit 7: PHY D, 0 = GMSL1, 1 = GMSL2. Bit 6: PHY C, 0 = GMSL1, 1 = GMSL2 Bit 5: PHY B, 0 = GMSL1, 1 = GMSL2 Bit 4: PHY A, 0 = GMSL1, 1 = GMSL2 Bit 3: PHY D, 0 = Link Disable, 1 = Link Enable Bit 2: PHY C, 0 = Link Disable, 1 = Link Enable Bit 1: PHY B, 0 = Link Disable, 1 = Link Enable Bit 0: PHY A, 0 = Link Disable, 1 = Link Enable

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customer under NDA

44.10.14.1.1 Programming Examples

The following examples demonstrate the configuration required to enable I2C address translation on both GMSL2 and GMSL1 links. Reference the I2C/UART section for serializer configuration information.

44.10.14.1.1.1 I2C Address Translation on GMSL2 (MAX9295A) Links

```
# Enable Link A only
0x52,0x0006,0xF1 // Turn on GMSL2 mode for link A
# Delay ~5ms
0x80,0x0000,0x82 // Change I2C address for this Link A serializer
0x82,0x0042,0xC4 // Set Ser source to 0xC4
0x82,0x0043,0x82 // Set Ser destination to 0x82
0x82,0x0044,0x22 // Set Image sensor source to 0x22
0x82,0x0045,0x20 // Set Image sensor destination to 0x20
# Enable Link B only
0x52,0x0006,0xF2 // Turn on GMSL2 mode for link B
# Delay ~5ms
0x80,0x0000,0x84 // Change I2C address for this Link B serializer
0x84,0x0042,0xC4 // Set Ser source to 0xC4
0x84,0x0043,0x84 // Set Ser destination to 0x84
0x84,0x0044,0x24 // Set Image sensor source to 0x24
0x84,0x0045,0x20 // Set Image sensor destination to 0x20
# Enable Link C only
0x52,0x0006,0xF4 // Turn on GMSL2 mode for link C
# Delay ~5ms
0x80,0x0000,0x86 // Change I2C address for this Link C serializer
0x86,0x0042,0xC4 // Set Ser source to 0xC4
0x86,0x0043,0x86 // Set Ser destination to 0x86
0x86,0x0044,0x26 // Set Image sensor source to 0x26
0x86,0x0045,0x20 // Set Image sensor destination to 0x20
# Enable Link D only
0x52,0x0006,0xF8 // Turn on GMSL2 mode for link D
# Delay ~5ms
0x80,0x0000,0x88 // Change I2C address for this Link D serializer
0x88,0x0042,0xC4 // Set Ser source to 0xC4
0x88,0x0043,0x88 // Set Ser destination to 0x88
0x88,0x0044,0x28 // Set Image sensor source to 0x28
0x88,0x0045,0x20 // Set Image sensor destination to 0x20
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute Only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.14.1.1.2 I2C Address Translation on GMSL1 (MAX96705) Links

```
# Enable Link A only
0x52,0x0006,0x01 // Turn on GMSL1 mode for link A
# Delay ~5ms
0x80,0x0000,0x82 // Change I2C address for this Link A serializer
0x82,0x0009,0xC4 // Set Ser source to 0xC4
0x82,0x000A,0x82 // Set Ser destination to 0x82
0x82,0x000B,0x22 // Set Image sensor source to 0x22
0x82,0x000C,0x20 // Set Image sensor destination to 0x20
# Enable Link B only
0x52,0x0006,0x02 // Turn on GMSL1 mode for link B
# Delay ~5ms
0x80,0x0000,0x84 // Change I2C address for this Link B serializer
0x84,0x0009,0xC4 // Set Ser source to 0xC4
0x84,0x000A,0x84 // Set Ser destination to 0x84
0x84,0x000B,0x24 // Set Image sensor source to 0x24
0x84,0x000C,0x20 // Set Image sensor destination to 0x20
# Enable Link C only
0x52,0x0006,0x04 // Turn on GMSL1 mode for link C
# Delay ~5ms
0x80,0x0000,0x86 // Change I2C address for this Link C serializer
0x86,0x0009,0xC4 // Set Ser source to 0xC4
0x86,0x000A,0x86 // Set Ser destination to 0x86
0x86,0x000B,0x26 // Set Image sensor source to 0x26
0x86,0x000C,0x20 // Set Image sensor destination to 0x20
# Enable Link D only
0x52,0x0006,0x08 // Turn on GMSL1 mode for link D
# Delay ~5ms
0x80,0x0000,0x88 // Change I2C address for this Link D serializer
0x88,0x0009,0xC4 // Set Ser source to 0xC4
0x88,0x000A,0x88 // Set Ser destination to 0x88
0x88,0x000B,0x28 // Set Image sensor source to 0x28
0x88,0x000C,0x20 // Set Image sensor destination to 0x20
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribution only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.15 Pass-Through SPI

The GMSL2 CSI-2 quad deserializers include two pass-through SPI blocks that can be used with GMSL2 serializers to bridge the SPI interface through the serial link (Figure 227). With SPI pass-through operation, an SPI master on one side of the GMSL2 link can control SPI peripherals on the opposite side.

Functionally, the GMSL2 link does not act as a transparent bridge. On the local side, an internal SPI slave receives data from an external SPI master and transmits it across the serial link. On the remote side, the device receives the data and uses an internal SPI master to transmit the data to the external SPI slave devices. Within the serial link, this has the effect of appearing as a 4-wire SPI master controlled by a 4-wire SPI slave. Reference the Serial Peripheral Interface (SPI) for a detailed explanation of SPI bridging. See Table 254 for GMSL2 CSI-2 quad deserializer SPI registers.

GMSL2 CSI-2 quad deserializer SPI ports do not support broadcasting and are intended for point-to-point communication. There are two SPI ports that can interface with two separate GMSL2 serializers via different links. Ports cannot be switched between links on the fly; the SPI engine must first be disabled before switching links.

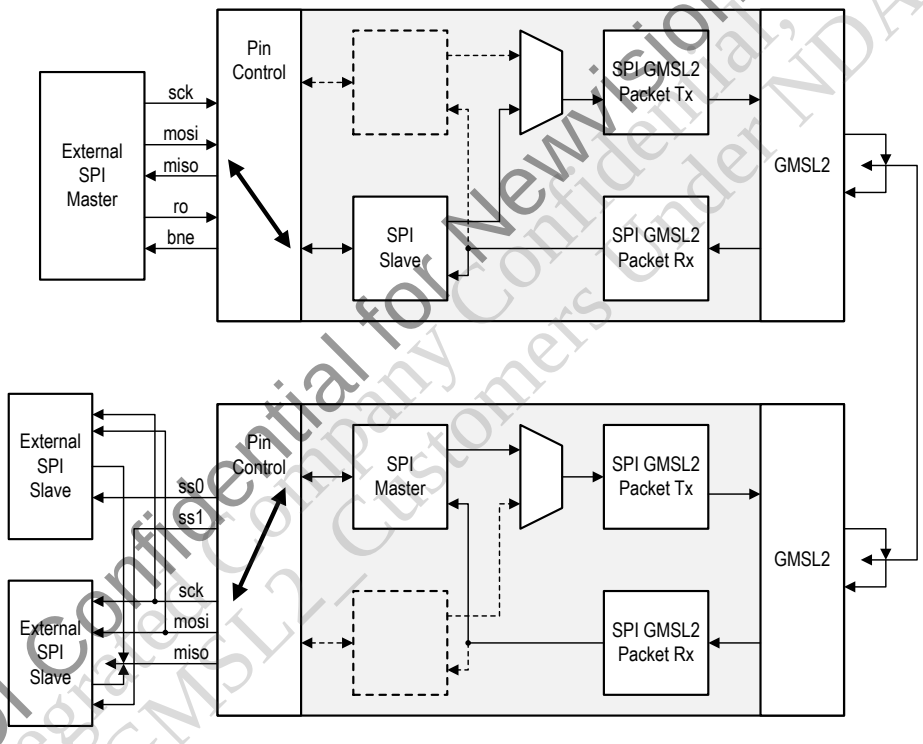


Figure 227. GMSL2 SPI Interface

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Maxim Integrated Company Confidential, Provided to GMSL2 Customers under NDA

Table 254. SPI Setting Registers

Register	Bits	Default Value	Description
0x04E0/0x04F0	7:0	0x08	**SPI0_0/SPI1_0 Register:** Bits [7:6]: Program to local ID if filtering packets based on header ID. 00 = ID 0 01 = ID 1 10 = ID 2 11 = ID 3 Bits [5:4]: ID for GMSL2 Header in SPI control channel bridge mode. 00 = ID 0 01 = ID 1 10 = ID 2 11 = ID 3 Bit 3: Select if SPI should ignore header ID to decide on packet. 0 = Accept only packets with proper ID. 1 = Ignore ID and accept all packets. Bit 2: Enable control channel SPI bridge function. 0 = Disabled 1 = Enabled Bit 1: Select if SPI is master or slave. 0 = SPI slave 1 = SPI master Bit 0: Enable SPI channel 0 = SPI channel disabled 1 = SPI channel enabled
0x04E1/0x04F1	7:2	000111	**SPI0_1/SPI1_1 Register:** Bits [7:6]: Sets the packet size ((2N+1) bytes) for GMSL2 SPI packets. If this is programmed to a value more than 7, ARQ of the SPI channel must be disabled. 000000: Packet size is 1 byte 000001: Packet size is 3 bytes … 111111: Packet size is 127 bytes
0x04E2/0x04F2	2:0	011	**SPI0_2/SPI1_2 Register:** Bit 2: Select SPI mode 0 or 3. 0 = SPI mode 0 1 = SPI mode 3 Bit 1: Set the polarity for SS2 when the SPI is a master. 0 = Slave select 2 is active low 1 = Slave select 2 is active high Bit 0: Set the polarity for SS1 when the SPI is a master. 0 = Slave select 1 is active low 1 = Slave select 1 is active high
0x04E3/0x04F3	7:0	0x00	**SPI0_3/SPI1_3 Register:** Bits [7:0]: Number of 300MHz clocks to delay between: 1. Assertion of SS and start of SCK pulses; 2. End of SCK pulses and de-assertion of SS; 3. De-assertion of SS and re-assertion of SS (if necessary).
0x04E4/0x04F4	7:0	0x00	**SPI0_4/SPI1_4 Register:** Bits [7:0]: Number of 300MHz clocks of SCK low time.

ADI Confidential for review under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers under NDA

0x04E5/0x04F5	7:0	0x00	**SPI0_5/SPI1_5 Register:** Bits [7:0]: Number of 300MHz clocks of SCK high time.
0x04E6/0x04F6	5:0	000000	**SPI0_6/SPI1_6 Register:** Bit 5: Alternate BNE status read back bit if GPIO is not available. 0 = Buffer empty 1 = Buffer not empty Bit 4: Alternate R/W control if GPIO is not available. 0 = Write 1 = Read Bit 3: Enable GPIO for use as slave select 2 output. 0 = Not used 1 = Used Bit 2: Enable GPIO for use as slave select 1 output. 0 = Not used 1 = Used Bit 1: Enable GPIO for use as BNE output for SPI data available status. 0 = Not used 1 = Used Bit 0: Enable GPIO for use as RO input for control of SPI data R/W. 0 = Not used 1 = Used
0x04E9/0x04F9	1:0	00/01	**SPI0_9/SPI1_9 Register:** Bits [1:0]: Select GMSL link connection. 00 = GMSL2 link A 01 = GMSL2 link B 10 = GMSL2 link C 11 = GMSL2 link D

ADI Confidential for Review under NDA
Analog Devices Confidential,
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.16 MIPI D-PHY Deskew Settings

In compliance with the MIPI D-PHY v1.2 specification, GMSL2 CSI-2 quad deserializer MIPI D-PHY ports support mandatory deskew calibration upon initialization. Additionally, the CSI-2 quad deserializers support optional periodic deskew as described by the MIPI Alliance. The D-PHY deskew mechanism is only relevant to lane speeds greater than 1.5Gbps per lane.

Note: Periodic deskew requires a continuous clock and that the clock lane is always in high-speed mode.

The initial transmit deskew is set with DESKEW_INIT[7:0] in registers 0x0903/0x0943/0x0983/0x09C3, which correspond to each MIPI PHY/Controller. All settings must be configured before video streams are received. After video lock, the MIPI Tx clock lane initiates automatically and an automatic initial deskew pattern is generated before the first HS data transmission. For system flexibility, an additional initial deskew pattern can be inserted by changing DESKEW_INIT bit 5 any time after the clock lane has been enabled. Note that DESKEW_INIT bit 4 must be set in order to trigger manual deskew.

The periodic deskew is initiated by setting bit 7 of DESKEW_PER[7:0] in registers 0x0904/0x0944/0x0984/0x09C4, which correspond to each MIPI PHY/Controller. The period interval has a programmable range from every 1 to 128 frames.

Register 0x06DF contains individual video pipe and MIPI controller resets (*Table 256*).

Table 255 contains MIPI D-PHY deskew registers. Refer to MIPI PHY Settings and the MIPI D-PHY Deskew section for additional information.

Table 255. MIPI D-PHY Deskew Registers

Register	Bits	Default Value	Description
0x0903	7:0	0x00	**MIPI TX 0 DESKEW_INIT Register:** Bit 7: Auto initial deskew on/off Bit 6: Reserved Bit 5: when bit 4 = 1, any change of this bit will trigger one-time immediate initial skew Bit 4: Manual initial on/off Bit 3: Reserved Bits [2:0]: Select initial deskew width: 1, 2, 3, … 8 * (32K) UI
0x0904	7:0	0x00	**MIPI TX 0 DESKEW_PER Register:** Bit 7: Period deskew calibration on/off Bit 6: Select generation on rising or falling edge of VS Bit [5:3]: Select periodic interval at every: 1, 2, 4, 8, … 128 frames Bit [2:0]: Select periodic deskew width: 1, 2, 3, … 8 * (1K) UI
0x0943	7:0	0x00	**MIPI TX 1 DESKEW_INIT Register:** Bit 7: Auto initial deskew on/off Bit 6: Reserved Bit 5: when bit 4 = 1, any change of this bit will trigger one-time immediate initial skew. Bit 4: Manual initial on/off Bit 3: Reserved

ADI Confidential for New vision under NDA
Analog Devices Confidential, Distribution only Under NDA
Maxim Integrated Confidential, Provided to GMSL 2 Customers Under NDA

			Bits [2:0]: Initial Deskew width 1, 2, … 8*(32K) UI
0x0944	7:0	0x00	**MIPI TX 1 DESKEW_PER Register:** Bit 7: Period deskew calibration on/off Bit 6: Select generation on rising or falling edge of VS Bit [5:3]: Select periodic interval at every: 1, 2, 4, 8, … 128 frames Bit [2:0]: Select periodic deskew width: 1, 2, 3, … 8 * (1K) UI
0x0983	7:0	0x00	**MIPI TX 2 DESKEW_INIT Register:** Bit 7: Auto initial deskew on/off Bit 6: Reserved Bit 5: when bit 4 = 1, any change of this bit will trigger one-time immediate initial skew. Bit 4: Manual initial on/off Bit 3: Reserved Bits [2:0]: Initial Deskew width 1, 2, … 8*(32K) UI
0x0984	7:0	0x00	**MIPI TX 2 DESKEW_PER Register:** Bit 7: Period deskew calibration on/off Bit 6: Select generation on rising or falling edge of VS Bit [5:3]: Select periodic interval at every: 1, 2, 4, 8, … 128 frames Bit [2:0]: Select periodic deskew width: 1, 2, 3, … 8 * (1K) UI
0x09C3	7:0	0x00	**MIPI TX 3 DESKEW_INIT Register:** Bit 7: Auto initial deskew on/off Bit 6: Reserved Bit 5: when bit 4 = 1, any change of this bit will trigger one-time immediate initial skew. Bit 4: Manual initial on/off Bit 3: Reserved Bits [2:0]: Initial Deskew width 1, 2, … 8*(32K) UI
0x09C4	7:0	0x00	**MIPI TX 3 DESKEW_PER Register:** Bit 7: Period deskew calibration on/off Bit 6: Select generation on rising or falling edge of VS Bit [5:3]: Select periodic interval at every: 1, 2, 4, 8, … 128 frames Bit [2:0]: Select periodic deskew width: 1, 2, 3, … 8 * (1K) UI

Table 256. Video Pipe and MIPI Controller Reset Registers

Register	Bits	Default Value	Description
0x06DF	0	0	**DP_RST_CC** Selects reset mode for iic_mux_uart blocks. 0: All I2C/UART ports will be reset during any one-shot reset or link reset. 1: Each I2C/UART port will be reset individually based on the associated GMSL PHY reset during a one-shot reset or link reset.
0x06DF	1	0	**DP_RST_FS** Selects reset mode to frame sync block. 0: Frame sync block will be reset during any one-shot reset or link reset. 1: The internal frame sync block will not be reset during a one-shot reset or link reset.

Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA
AD Confidential for New Version under NDA

0x06DF	2	0	**DP_RST_VP** Selects RST mode to video_rx, vrx blocks. 0: All video pipes will be reset during any one-shot reset or link reset. 1: Each video pipe will be reset individually based on the associated GMSL PHY reset during a one-shot reset or link reset.
0x06DF	3	0	**DP_RST_MIPI** Selects RST mode to MIPI controllers. 0: All MIPI controllers will be reset during any one-shot reset or link reset. 1: Each MIPI controller can be reset individually based on RST_MIPITX_LOC[3:0] (0x8C9).
0x06DF	4	0	**DP_RST_MIPI2** Selects RST mode to MIPI controllers. 0: All MIPI controllers will be reset during any one-shot reset or link reset. 1: Each MIPI controller can be reset automatically based on the associated GMSL PHY and Video Pipe reset during a one-shot reset or link reset.
0x06DF	5	1	**DP_RST_STABLE** Selects RST mode. 0: Do not prevent reset glitches when changing reset behavior between individualized resets (Rev C behavior) vs. non-individualized resets (Rev B behavior). 1: Prevent reset glitches when changing reset behavior between individualized resets (Rev C behavior) vs. non-individualized resets (Rev B behavior).
0x06DF	6	1	**DP_RST_MIPI3** Selects RST mode CMD FIFO read pointer. 0: CMD FIFO read pointer is not reset automatically when video lock is lost. 1: CMD FIFO read pointer is reset automatically when video lock is lost.

44.10.16.1.1 Programming Example

The following programming example shows a typical use of initial and periodic deskew calibration.

```
# Enable initial deskew packets with the minimum width 32K UI on controller 1 for port A
0x52,0x0943,0x80
# Enable periodic deskew packets with width 2K UI every 2 frames
0x52,0x0944,0x91
```

ADI Confidential for review under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL2 Customers Under NDA

44.10.17 MIPI C-PHY Settings

The GMSL2 CSI-2 quad deserializers are fully compliant with the MIPI C-PHY v1.0 standard. This physical interface uses three-phase symbol encoding that delivers 2.28 bits per symbol over three-wire trios operating at 2.5GSym/s per trio. Four trios have a peak bandwidth of 22.8Gbps. C-PHY and D-PHY are intended to be able to share device hardware; refer to MIPI specifications for design rules to ensure ideal system performance.

The GMSL2 CSI-2 quad deserializers support 2- and 4-lane C-PHY modes; further programming allows 1- and 3-lane C-PHY arrangements. Note that this is similar to the D-PHY configurations. C-PHY mode is enabled by setting CSI2_CPHY_EN = 1 (bit 5 in registers 0x090A/0x094A/0x098A/0x09CA). Each PHY is individually configured with a dedicated register. Refer to MIPI PHY Settings for more information.

According to the C-PHY v1.0 protocol, HS data transmissions have two programmable fields that may be adjusted to ensure that the C-PHY Rx has a more stable capture (Figure). The $t_{3\text{-PREBEGIN}}$ field of the preamble comprises multiple groups of seven “3” symbols. The purpose of the $t_{3\text{-PREBEGIN}}$ field is to provide a sufficient number of clocks to the upper layer protocol so that pipeline stages are initialized prior to receiving data. Similarly, the $t_{3\text{-POST}}$ field at the end of the packet data is identified by a unique sequence of “4” symbols. See Table 257 for relevant timing registers.

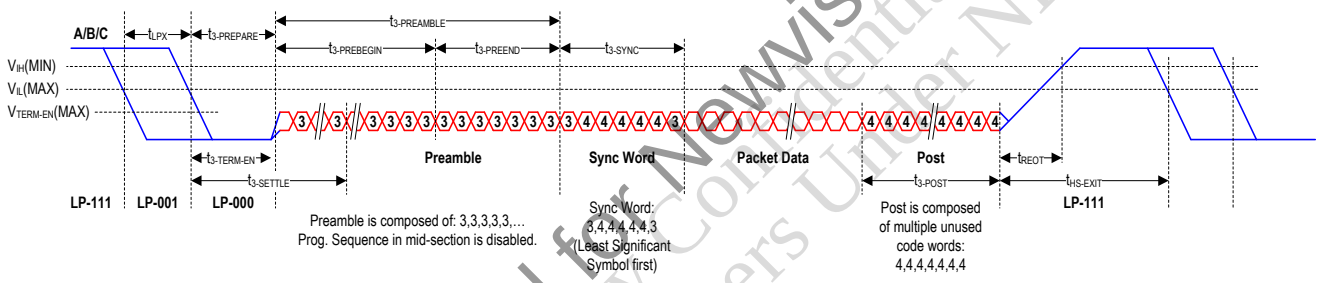


Figure 20. C-PHY v1.0 HS Burst Data Transmission

Table 257. MIPI C-PHY Timing Setting Registers

Register	Bits	Default Value	Description
0x08AD	5:0	000000	**$t_{3\text{-PREBEGIN}}$ Register:** C-PHY Pre-begin phase of the preamble length = ($t_{3\text{-PREBEGIN}}$ + 1) * 7 UI.
0x08AE	6:2	00000	**$t_{3\text{-POST}}$ Register:** C-PHY Post length = ($t_{3\text{-POST}}$ + 1) * 7 UI.

44.10.17.1.1 Programming Example

```
# Set 4 lane C-PHY for MIPI PHY 1
0x52,0x094A,0xE0
# C-PHY timing adjustment
0x52,0x08AD,0x1F
0x52,0x08AE,0x5C
```

MAXIM Confidential for New vision under NDA
Maxim Confidential, Distributed Company Confidential, Provided for MSI_2_Customers Under NDA

44.10.18 Sync Pulse Output

The GMSL2 CSI-2 quad deserializers are capable of outputting the VS, HS, and DE signals from the video pipe data streams.

All three pulses can be output at the same time on three different GPIOs.

- MFP4 – VS output GPIO
- MFP13 – HS output GPIO
- MFP14 – DE output GPIO

GPIOs generally multiplex with multiple functions. Therefore, to output sync pulses, the default or higher-priority GPIO function(s) of the above pins must be disabled. In particular, the MFP4/MFP14 pins are enabled as the third I2C/UART pins by default. MFP4 is also used as a GPI pin in GMSL1 mode. Refer to the I/O Matrix (MFP) section for additional details.

Reference Table 258 for sync pulse output registers.

Table 258. Sync Pulse Output Registers

Register	Bits	Default Value	Description
0x00FA	7:0	0x00	**HVD_GPIO_CTRL register:** Bit 6: Enables DE out to GPIO Bit 5: Enables HS out to GPIO Bit 4: Enables VS out to GPIO Bit [2:0]: Selects the pipes to view VS, HS, and/or DE 000 = Select pipe 0 001 = Select pipe 1 010 = Select pipe 2 011 = Select pipe 3 100 = Select pipe 4 101 = Select pipe 5 110 = Select pipe 6 111 = Select pipe 7

ADI Confidential for New Version under NDA
Analog Devices Confidential Information subject to NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers_Only_under_NDA

Alternatively, sync pulse signals can be monitored via registers. Registers 0x11F0/0x11F1/0x11F2 contain status flags indicating the detection of DE/HS/VS signals for each video pipe. The polarity of the HS/VS signals is accessed with registers 0x11F3/0x11F4. See Table 259 for additional information.

Table 259. Sync Signal Status Registers

Register	Bits	Default Value	Description
0x11F0	7:0	N/A	**DE_DET registers:** Bits 0 through 7 correspond with video pipes 0 through 7. Bitfield name is appended with video pipe number. Indicates if DE signal is detected in video pipe. 0b0: DE is not detected 0b1: DE is detected
0x11F1	7:0	N/A	**HS_DET registers:** Bits 0 through 7 correspond with video pipes 0 through 7. Bitfield name is appended with video pipe number. Indicates if HS signal is detected in video pipe. 0b0: HS is not detected 0b1: HS is detected
0x11F2	7:0	N/A	**VS_DET registers:** Bits 0 through 7 correspond with video pipes 0 through 7. Bitfield name is appended with video pipe number. Indicates if VS signal is detected in video pipe. 0b0: VS is not detected 0b1: VS is detected
0x11F3	7:0	N/A	**HS_POL registers:** Bits 0 through 7 correspond with video pipes 0 through 7. Bitfield name is appended with video pipe number. Indicates the detected HS signal polarity in video pipe. 0b0: Active low 0b1: Active high
0x11F4	7:0	N/A	**VS_POL registers:** Bits 0 through 7 correspond with video pipes 0 through 7. Bitfield name is appended with video pipe number. Indicates the detected VS signal polarity in video pipe. 0b0: Active low 0b1: Active high

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential, Provided to GMSL2 Customers Under NDA

44.10.18.1.1 Programming Examples

The following examples demonstrate the configuration required to output sync pulses in both GMSL2 and GMSL1 modes.

44.10.18.1.1.1 Output pipe 1 VS/HS on MFP4/MFP13 (GMSL2 Mode)

```
# Disable the 3rd I2C/UART control channel (set bit 2) to release MFP4
0x52,0x0001,0xCC
# Enable pipe 1 VSYNC/HSYNC output
0x52,0x00FA,0x31
```

44.10.18.1.1.2 Output pipe 0 VS on MFP4 (GMSL1 Mode)

```
# Disable the 3rd I2C/UART control channel (set bit 2) to release MFP4
0x52,0x0001,0xCC
# Disable GPI functions from all 4 links to release MFP4
0x52,0x0B08,0x01
0x52,0x0C08,0x01
0x52,0x0D08,0x01
0x52,0x0E08,0x01
# Enable pipe 0 VSYNC output
0x52,0x00FA,0x10
```

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

44.10.19 Video Mask Output

GMSL2 CSI-2 quad deserializers can automatically mask video pipe outputs if a video pipe's video lock is lost. When enabled, zeroes are automatically inserted into the synchronized aggregated video output(s) of a video pipe that has lost video lock. This allows the other video streams to continue being transmitted uninterrupted on the MIPI interface. Video pipes can also be manually forced to mask video output streams regardless of video lock status. Reference Table 260 for configuration details.

In Figure 228, the video stream output from three cameras is aggregated via side-by-side concatenation. In Figure 229, the 'Camera 1' video pipe output is masked while the video data from the other two cameras continues to be transmitted.



Figure 228. Concatenated Video Stream from Three Cameras



Figure 229. Concatenated Video Stream with Masked Video

ADI Confidential for New Revision under NDA
Analog Devices Confidential
Maxim Integrated Company Confidential
Provided to GMSL Customers Under NDA

Table 260. Video Mask Output Registers

Register	Bits	Default Value	Description
0x08C6	3:0	0x0000	**Auto_Mask_En register:** Automatically insert 0's into synchronized aggregated video outputs if a Video Pipe 0-3 video lock is lost. This allows the other video streams to continue being transmitted on the MIPI interface. 0bXXX1: Auto video mask enabled for Video Pipe 0 0bXX1X: Auto video mask enabled for Video Pipe 1 0bX1XX: Auto video mask enabled for Video Pipe 2 0b1XXX: Auto video mask enabled for Video Pipe 3
0x08C6	7:4	0x0000	**Force_Video_Mask register:** Forces video output to be masked (send all 0's) in 4WxH or Wx4H synchronous aggregation modes. Masking impacts video streams from video pipes 0-3. 0bXXX1: Force video from Video Pipe 0 to be masked. 0bXX1X: Force video from Video Pipe 1 to be masked. 0bX1XX: Force video from Video Pipe 2 to be masked. 0b1XXX: Force video from Video Pipe 3 to be masked.

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution Only
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.10.20 Error-Correcting Code (ECC) Memory

GMSL2 CSI-2 quad deserializers integrate error-correcting code (ECC) memory to protect against undetected pixel data corruption within the video pipes. ECC enables the correction of 1-bit errors and detection of 1- or 2-bit errors per 32 bits of video data. This functionality provides protection of video data between the CRC protection of the incoming GMSL2 data stream and the outgoing CSI-2 data stream, ensuring that internally generated silent bit errors do not corrupt the outgoing CSI-2 data.

ECC memory thresholds are configured with the MEM_ECC_ERR1_THR and MEM_ECC_ERR2_THR bitfields in register 0x1250. The EEC memory error counts are reset with RESET_MEM_ECC_ERR1_CNT and RESET_MEM_ECC_ERR2_CNT bitfields. Intentional memory error injection is configured with the MEM_INJ_E1A and MEM_INJ_E2A bitfields of register 0x1253. Configuration details are provided in Table 261 below.

Table 261. Error-Correcting Code (ECC) Memory Configuration Registers

Register	Bits	Default Value	Description
0x1250	0	0x0	**RESET_MEM_ECC_ERR1_CNT register:** Reset memory ECC 1-bit error count to 0. 0b0: Do not reset ECC 1-bit error count 0b1: Reset ECC 1-bit error count
0x1250	1	0x0	**RESET_MEM_ECC_ERR2_CNT register:** Reset memory ECC 2-bit error count to 0. 0b0: Do not reset ECC 2-bit error count 0b1: Reset ECC 2-bit error count
0x1250	4:2	0x000	**MEM_ECC_ERR1_THR register:** Decoding and error reporting threshold. MEM_ECC_ERR1 is asserted when MEM_ECC_ERR1_CNT ≥ MEM_ECC_ERR1_THR. 0b000: 1 0b001: 2 0b010: 4 0b011: 8 0b100: 16 0b101: 32 0b110: 64 0b111: 128
0x1250	7:5	0x000	**MEM_ECC_ERR2_THR register:** Decoding and error reporting threshold. MEM_ECC_ERR2 is asserted when MEM_ECC_ERR2_CNT ≥ MEM_ECC_ERR2_THR. 0b000: 1 0b001: 2 0b010: 4 0b011: 8 0b100: 16 0b101: 32 0b110: 64 0b111: 128
0x1253	0	0x0	**MEM_INJ_E1A register:**

ADI Confidential for Review under NDA
Analog Devices Integrated Company Confidential
Maxim Integrated Company Confidential
Provided to GMSL2_Customers under NDA

			Injects 1 error bit for every 39-bit write into all 64 ECC memories in backtop. 0b0: Do not inject error bits 0b1: Inject error bits
0x1253	0	0x0	**MEM_INJ_E2A register:** Injects 2 error bits for every 39-bit write into all 64 ECC memories in backtop. 0b0: Do not inject error bits 0b1: Inject error bits

ADI Confidential for Newvision under NDA
Analog Devices Confidential. Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

44.10.21 MIPI Packet Count

The MIPI packet count registers are used to determine whether MIDP data is flowing in the GMSL2 CSI-2 quad deserializer. The csi2_tx*_pkt_cnt registers report the packet count of the associated MIPI controller; the phy*_pkt_cnt registers report the packet count of the associated MIPI PHY. Sequential reads returning different values indicate that data is being transmitted by the associated MIPI controller or PHY. The register value will not change if data is not flowing. Reference Table 262 for MIPI packet count registers.

Table 262. MIPI Packet Count Registers

Register	Bits	Default Value	Description	Decode
0x08D0	3:0	0x00	Packet count of CSI-2 Controller 0	**csi2_tx*_pkt_cnt registers:** 0bXXX1: PHY copy 0 FIFO overflow 0bXX1X: PHY copy 0 FIFO underflow 0bX1XX: PHY copy 1 FIFO overflow 0b1XXX: PHY copy 1 FIFO underflow
0x08D0	7:4	0x00	Packet count of CSI-2 Controller 1	
0x08D1	3:0	0x00	Packet count of CSI-2 Controller 2	
0x08D1	7:4	0x00	Packet count of CSI-2 Controller 3	
0x08D2	3:0	0x00	Packet count of MIPI PHY0	**phy*_pkt_cnt registers:** 0bXXX1: PHY copy 0 FIFO overflow 0bXX1X: PHY copy 0 FIFO underflow 0bX1XX: PHY copy 1 FIFO overflow 0b1XXX: PHY copy 1 FIFO underflow
0x08D2	7:4	0x00	Packet count of MIPI PHY1	
0x08D3	3:0	0x00	Packet count of MIPI PHY2	
0x08D3	7:4	0x00	Packet count of MIPI PHY3	

ADI Confidential for NewVision under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

44.11 Feature Availability by Device

The following table details the feature availability for the GMSL2 CSI-2 Quad Deserializers (Table 173).

Table 263. CSI-2 Quad Deserializer Device Family Part Numbers

Device Number	Forward Link Speed (Maximum)
MAX96712	6Gbps
MAX96712B	6Gbps
MAX96722	3Gbps
MAX96722B	3Gbps

Note: The MAX96712/MAX96712B and the MAX96722/MAX96722B parts are functionally equivalent except for the inclusion of Reference over Reverse (RoR) in the B-type device variants in GMSL2 mode.

ADI Confidential for Newvision under NDA
Analog Devices Confidential Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

45 MIPI D-PHY Deskew

45.1 Overview

45.2 GMSL2 DSI and CSI-2 devices integrate MIPI D-PHY v1.2 ports compliant with the specifications published by the MIPI Alliance. For these devices, serializers have D-PHY input ports, and deserializers have D-PHY output ports. Reference the Debug

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.
- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

45.2.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

ADI Confidential for New Vision under NDA
Analog Devices Confidential
Maxim Integrated Confidential
Provided to GMSL2 Company Confidential Customers Under NDA

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.
Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock,

ADI Confidential Distribution Only Under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customer Under NDA

					lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

45.2.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a

ADI Confidential for Newvision under NDA
Maxim Integrated Confidential
Provided to GMSL2_Customers Under NDA

		watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for NewVision Under NDA
Analog Devices Confidential Distribution
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

45.3 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 and DSI Serializers sections for more information.

MIPI D-PHY v1.2 supports individual lanes speeds up to 2.5Gbps per lane. To provide the most robust and reliable MIPI connection, deskew calibration is available to minimize the Tx-to-Rx MIPI clock-to-data skew. In a GMSL2 system, this can occur in two different locations: from the MIPI source device to the GMSL2 serializer and/or from the GMSL2 deserializer to the MIPI sink device. Links available for MIPI D-PHY Deskew are indicated in Figure 230 below (i.e., 'MIPI Link 1' and 'MIPI Link 2').

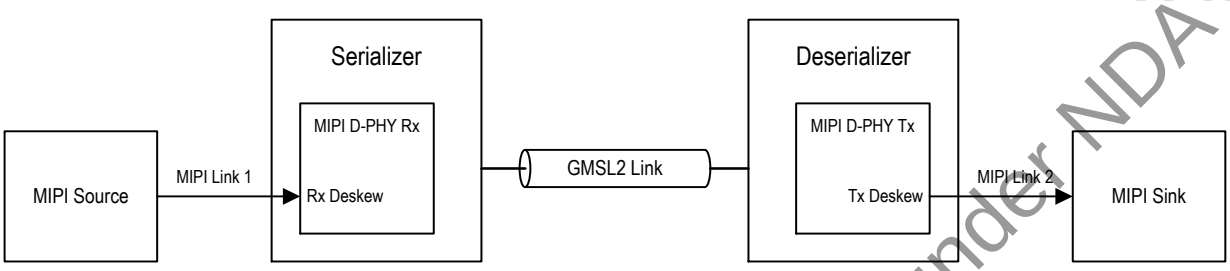


Figure 230. MIPI D-PHY Deskew Within a GMSL2 System

Note: The combination of MIPI Tx pins to the MIPI Rx pins must be within the MIPI deskew guidelines for the deskew procedure to be valid. GMSL2 D-PHY devices only support continuous clock mode.

Deskew calibration is intended for D-PHY lane speeds in excess of 1.5Gbps.

The MIPI D-PHY receiver in the serializers supports the deskew calibration. When it is enabled, the D-PHY receiver detects a special deskew burst from the MIPI transmitter. The D-PHY receiver then uses a deskew pattern to internally align the clock and data lanes. This alignment process increases clock-to-data skew tolerance and reduces data errors. The MIPI D-PHY transmitter in the deserializers sends out the special deskew packets if deskew calibration is required.

GMSL2 MIPI D-PHY devices, in compliance with the D-PHY v1.2 specification, support both the mandatory deskew calibration upon initialization and the optional periodic deskew calibration. The transmission of the deskew calibration sequence is not required for D-PHY lane speeds below 1.5Gbps; periodic deskew is optional for all lane speeds.

*45.4 **Note**: GMSL2 MIPI D-PHY v1.2 input and output ports are compliant with published MIPI Alliance specifications. Reference appropriate documentation for more information regarding the D-PHY v1.2 specification. Reference the Debug*

The following steps can be followed to diagnose a problem if there is no video output on the display. Detailed debug/diagnostic register information is included at the end of this section.

- **Check for GMSL2 link lock.** Read LOCKED status bit or look at LOCK pin status.
- **Check for valid input video.** Read serializer bit PCLKDET and ensure it is high for the video pipe that is receiving video.

ADI Confidential for review under NDA
Analog Devices Confidential Distributed to Maxim Integrated Company Confidential Provided to GMSL2_Customers under NDA

- **Check for video lock across the GMSL2 link.** Read deserializer register 0x01DC and check the bit VIDEO_LOCK. If this is low, then video is not being transferred across the link correctly.
- **Check detected pixel clock (PCLK) in the deserializer.**
  When the VTRG is enabled, the detected PCLK in kHz can be read from PB_CLKP_ADJ_STATUS_B0, PB_CLKP_ADJ_STATUS_B1, and PB_CLKP_ADJ_STATUS_B2
  This is a very precise indicator of the PCLK of the received video stream.
- **Check for successful DP link training.**
  - Link training can be checked by reading register 0x7F0/0x7F1 or by reading the link training status registers in the sink DPCD.
  - If link training was successful and there is still no video, check that the resolution and blanking of the input video at the serializer perfectly matches the expected timing programmed into the DP Video Configuration Registers.
  - Check that the DP link parameter (link rate and lane count) register values are configured correctly and match the sink device capability.
- **Debug the eDP/DP Controller state machine.** Check the controller status/error indicators (Table 199) and diagnostic indicators (Table 200).
- **Check for hardware issue(s).** If there is still no video after completing the previous steps, the eDP/DP sink device may be incompatible, or there may be a hardware issue compromising signal integrity on the eDP/DP link.

45.4.1 DP Link Training Status Registers

DP controller status information is contained in two primary registers. The registers include indicators for common errors and status change events. Each status indicator bit can be individually masked to enable selection of which specific event indicators drive the global PMX_ERR_FLAG bit, which can generate an interrupt at the ERRB pin if desired. DP Status indicators can be found in registers 0x7F0 and 0x7F1 while the corresponding mask bits are located in registers 0xE7AE and 0xE7AF.

In general, the 0x7F0/0x7F1 status registers hold the most recent singular status change event. These registers do not hold a comprehensive log of prior event indications. Each time any event occurs, the status registers are updated to reflect the latest event only.

0x7F1[7] is a special global flag bit that is set any time an unmasked event occurs. When this bit is set, two important things happen. The first is that the PMX_ERR_FLAG is set. The second is that registers 0x7F0 and 0x7F1 are locked. They can no longer be updated by the DP controller until the user clears 0x7F1[7] by reading register 0x7F1. Other bits in registers 0x7F0 and 0x7F1 are not cleared by a user read and will only be cleared by the DP controller itself updating the status.

By default, all status event indicators are masked with the exception of the watchdog timer event, which cannot be masked. When an event indicator is masked, the corresponding bit in 0xE7AE/0xE7AF is set to 0, which is the default.

If some events are masked and others are unmasked, then the behavior of the status reporting system can be summarized as:

if 0x7F1[7] = 1:
First unmasked singular event since last read of 0x7F1[7] is reported in 0x7F1/0x7F0.

Maxim Integrated Confidential, Provided to GMSL Customers under NDA
ADI Confidential for new vision under NDA

Status registers have been locked, and a read operation will unlock the status registers.
if 0x7F1[7] = 0:
Most recent singular masked event (or first unmasked singular event if another status change event has not occurred since 0x7F1 was last read).

All available status indicators and associated mask bit locations are detailed in the following table. For register 0x7F0 status indicators, the associated mask bits are in register 0xE7AE at the same index as a given status indicator flag. Similarly, for register 0x7F1 status indicators, associated mask bits are in register 0xE7AF at the same index as a given status indicator bit. Note that any time an unmasked event occurs, 0x7F1[7] is set. This is why each status state corresponds to two possible values of register 0x7F1 depending on whether or not a given status event is masked. The 0x7F1[7] global error indicator that drives PMX_ERR_FLAG cannot be masked. Any time it is set, the PMX_ERR_FLAG asserted.

Table 199. DP Link Training Status and Error Register Details

Register 0x07F0 Value	Register 0x07F1 Value	Status	Mask Bit	Message Type	Description
0x00	0x00	NO_TRAIN_ERR	0xE7AE[0]	Status	No Link training occurred
0x01	0x00/0x80	TRAIN_SUCCESSFUL	0xE7AE[0]	Status	Training was successful
0x00	0x40/0xC0	TP1_TRAIN_ERR	0xE7AF[6]	Error	Training Phase 1 failed. (Clock recovery failure)
0x00	0x20/0xA0	TP2_TRAIN_ERR	0xE7AF[5]	Error	Training Phase 2 failed. (Equalization, Symbol lock, lane alignment failure)
0x00	0x10/0x90	TRAIN_PARMS_ERR	0xE7AF[4]	Error	No training parameters found when link training was attempted
0x00	0x08/0x88	TRAIN_NOHPD_ERR	0xE7AF[3]	Error	HPD was low when training attempted
0x00	0x04/0x84	TRAIN_NOVL_ERR	0xE7AF[2]	Error	Video lock was low when training attempted
0x00	0x02/0x82	WATCHDOG_TRIGGER_ERR	0xE7AF[1]	Error	Watch Dog timer triggered
0x00	0x01/0x81	VL_LOST_ERR	0xE7AF[0]	Error	Video lock lost unexpectedly after link

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

					training completed
0x20	0x00/0x80	HPD_DISCONNECT	0xE7AE[5]	Status	HPD 1-> 0 disconnect event has occurred
0x80	0x00/0x80	HPD_IRQ_EVENT	0xE7AE[7]	Status	HPD Event or HPD IRQ occurred

45.4.2 DP Controller Diagnostic Registers

The DP controller includes additional diagnostic registers that provide further insight into how the controller observes relevant system behavior. This information may be useful for system debug.

Table 200. eDP/DP Controller State Machine Debug Registers

Register Address	Bitfield	Description
0xE75A 0xE75B	WATCHDOGCNT[7:0]	Contains the count of how many times the watchdog timer has triggered since power-up. The watchdog timer is set to a 1 second trigger. The watchdog timer should never trigger under normal operating conditions.
0xE75E	HPD_PIN_STATE[0]	Value of DP HPD pin. HPD must be logic high during normal operation.
0xE760	VL_STATE[0]	Video channel is locked and outputting valid video data. VL_STATE must be high for link training to occur.
0xE762 0xE763	VL_LOST_CNT[15:0]	Will increment whenever video lock transitions from a high to low state. This variable is cleared at power-up and also by a watchdog timer event. Can also be reset to 0 by writing 0 to these registers.
0xE764 0xE765	eHPD_IRQ_COUNT[15:0]	Contains the count of how many DP IRQ events have occurred since power-up. A DP IRQ event is caused by the DP sink device driving the HDP signal low for a period between 0.5ms and 1.0ms. Occurrence of a DP IRQ event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE766 0xE767	eHPD_EVENT_COUNT[15:0]	Contains the count of how many DP disconnect events have occurred since power-up. A DP disconnect event is caused by the DP sink device driving the HDP signal low for a period greater than 2.0ms. Occurrence of a DP disconnect event will automatically result in retraining of the eDP/DP link. This variable is cleared at power-up and also by a watchdog timer event.
0xE768 0xE769	NACK_DEFER_CNT[7:0]	Contains the count of how many AUX cycles have resulted in either a NACK or DEFER status. When this occurs, the AUX cycle is automatically restarted. This variable is cleared at power-up and also by a watchdog timer event.

ADI Confidential for New Vision under NDA
Analog Confidential Content
Maxim Integrated Confidential, Provided under NDA
Provided to GMSL2_Customer Under NDA

45.5 Feature Availability by Device

eDP/DP is only available on the eDP/DP deserializer device family (Table 201).

Table 201. eDP/DP Deserializer Device Family Part Numbers

Device Family	Part Number
eDP/DP Deserializers	MAX96772
	MAX96774
	MAX96776
	MAX96778

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

CSI-2 and DSI Serializers sections for additional information.

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

45.6 Operation

GMSL D-PHY serializers support the Rx portion of the deskew calibration(s); D-PHY deserializers support the Tx portion. Deskew calibration is disabled by default. When operating the MIPI D-PHY above 1.5Gbps, deskew calibration must be enabled via the enable register bit.

Both the initial and periodic deskew calibrations are enabled through register writes. Note that these deskew procedures differ in length of the calibration sequence.

In the serializer, Rx deskew is enabled after configuring the data rate, lane selection, and D-PHY lane map but prior to receiving video from the MIPI source. Once enabled, the initialization calibration sequence is automatically generated when the MIPI D-PHY receiver in the serializer detects the sync pattern from the source. The deskew calibration function has successfully aligned the clock and data lanes when the clock-to-data skew is less than ±0.4UI.

Note: The first packet received by the MIPI D-PHY receiver must be the deskew packet. Once the receiver detects the deskew sync pattern, the calibration sequence begins.

The Tx deskew registers must be set prior to configuring the CSI-2 PLL. After CSI-2 PLL lock and video lock are established, the MIPI Tx clock lane starts and is automatically followed by the initial deskew pattern. The Tx deskew burst uses a sync pattern comprising a series of ones for a duration of 16 UI. After the sync pattern is sent, a clock pattern payload is transmitted. This pattern comprises alternating zeroes and ones (i.e., 010101010101…) for a minimum duration of 2^{15} UI for the initial deskew calibration and 2^{10} UI for the periodic calibration(s).

Figure 231 below depicts a high-speed MIPI data transmission in normal mode. Figure 232 shows the deskew calibration process (see Figure 233 for a detailed view).

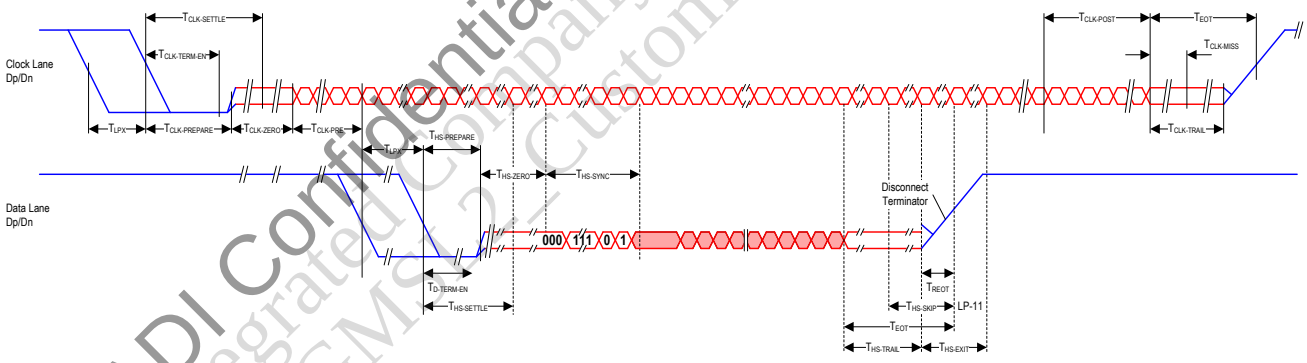


Figure 231. High-Speed Data Transmission in Normal Mode

ADI Confidential for New Version under NDA
Analog Devices Confidential Information
Maxim Integrated Company Confidential
Provided to GMSL 2 Customers Under NDA

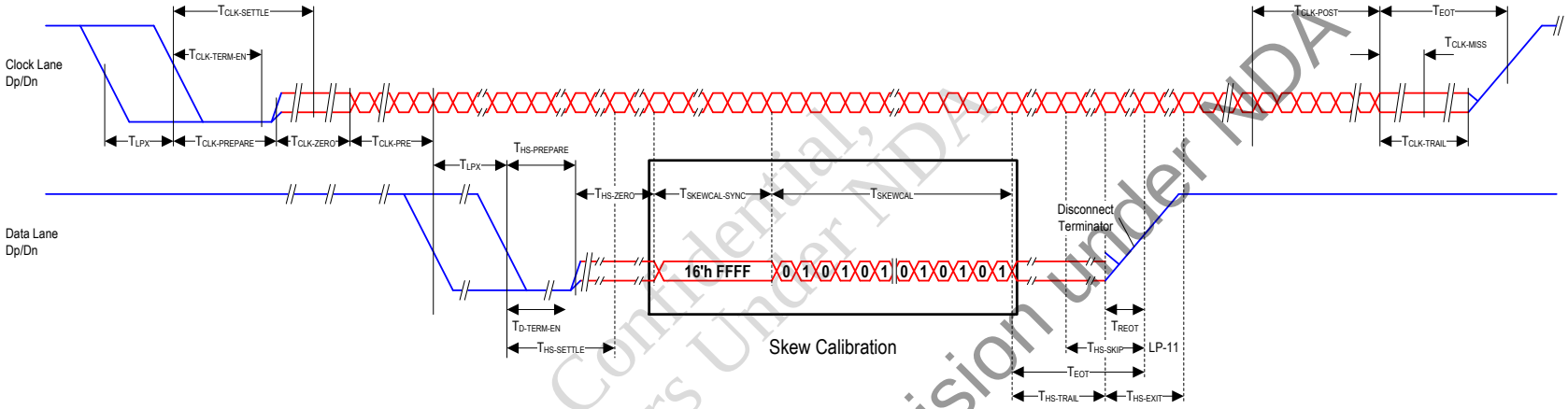


Figure 232. High-Speed Deskew Calibration

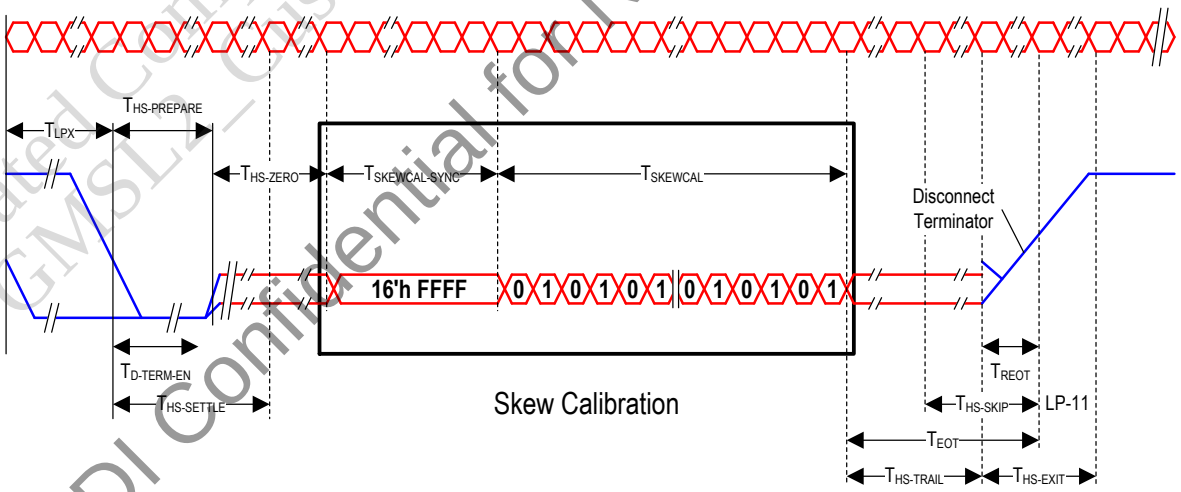


Figure 233. Skew Calibration (Detailed View)

Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA
ADI Confidential for Newvision under NDA

45.6.1 Modes of Operation

GMSL2 MIPI D-PHY deserializers have two modes of MIPI deskew packet generation: Initial Deskew and Periodic Deskew.

Initial Deskew

- **Auto Mode**: Initial deskew is automatically generated upon MIPI power-up when enabled.
- **Manual Mode**: Initial deskew is inserted once between MIPI packets via register control.

Periodic Deskew

- **Auto Mode**: The periodic deskew is automatically generated. The occurrence and length of the deskew are programmable.

The deskew modes differ by length of training sequence. The initial deskew training sequence is longer than that of the periodic deskew. The initial deskew sequence is longer because the receive needs sufficient time to calibrate the MIPI clock frequency. Periodic deskew is used to fine-tune the skew to compensate for operational variations in supply and temperature.

Note: GMSL2 MIPI D-PHY serializers are compatible with both initial and periodic deskew.

In the serializer, the Rx deskew is enabled after configuring the data rate, lane selection, and D-PHY lane map but prior to receiving video from the MIPI source. If the deskew is enabled, the first packet received by the MIPI receiver must be the initial deskew packet. In the deserializer, the Tx deskew registers must be set prior to configuring the CSI-2 PLL.

Note: The Rx deskew is calibrated to the current MIPI clock frequency. Therefore, the clock frequency must be stable before the deskew packet is sent out. The MIPI clock frequency must remain stable after the calibration.

ADI Confidential for New Version under NDA
Analog Devices Confidential, Published on Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

45.7 Configuration

PHY and controller availability is package-dependent; reference device-specific datasheet(s) for more information.

45.7.1 D-PHY Serializers

Deskew calibration for GMSL2 serializers with D-PHY is enabled via the MIPI_RX1 register. Each controller has a dedicated enable bit and is used for both initial and periodic deskew calibration:

- Controller 0: MIPI_RX1[2]
- Controller 1: MIPI_RX1[6]

The initial Rx deskew procedure must be completed before receiving MIPI packets. For periodic deskew, the Rx deskew enable be configured before the start of MIPI operations (including clock and data lanes). Deskew enable cannot be dynamically changed and must remain active after the initialization deskew procedure.

Each MIPI PHY has a dedicated deskew calibration status register. There are two status bits for each lane: one bit indicates whether the calibration pattern was received, and the other bit indicating if the calibration was successful. MIPI PHY lane configuration must be considered when monitoring the status registers.

Deskew calibration status registers:

- PHY0: MIPI_RX10
- PHY1: MIPI_RX12
- PHY2: MIPI_RX14
- PHY3: MIPI_RX16

Table 264. D-PHY Serializers Deskew Calibration Configuration Registers

Register	Bitfield	POR	Description
MIPI_RX1	ctrl0_deskewen ctrl1_deskewen	0 0	**Deskew calibration settings:** Bit 6: ctrl1_deskewen for port B • 0: Deskew calibration disabled • 1: Deskew calibration enabled Bit 2: ctrl0_deskewen for port A • 0: Deskew calibration disabled • 1: Deskew calibration enabled
MIPI_RX10 **MIPI_RX12** **MIPI_RX14** **MIPI_RX16**	phy0_hs_err[4:7] phy1_hs_err[4:7] phy2_hs_err[4:7] phy3_hs_err[4:7]	0x00	**Deskew calibration status:** Bit 7: Deskew calibration pattern flag on data lane 0 • 0: Not received • 1: Received Bit 6: Deskew calibration pattern flag on data lane 1 • 0: Not received • 1: Received Bit 5: Deskew calibration failure flag on data lane 0 • 0: Default • 1: Failed Bit 4: Deskew calibration failure flag on data lane 1 • 0: Default • 1: Failed

ADI Confidential for NewVision under NDA
Maxim Integrated Confidential Distribution Only
Provided to GMSL2_Customers Under NDA
Company Confidential

45.7.2 D-PHY Deserializers

The initial deskew pattern for GMSL2 D-PHY deserializers is set with the DESKEW_INIT registers. Note that each controller is independently configured. These registers must be programmed before configuring the CSI-2 PLL settings (i.e., the MIPI clock and data rates).

Automatic initial deskew is enabled by writing DESKEW_INIT[7]. The MIPI Tx clock lane will start after video lock and CSI-2 PLL lock are established, followed by the transmission of an automatic initial deskew pattern. At any point after the clock lane is established, a one-time initial deskew pattern can be inserted. Manual initial deskew generation is configured with the DESKEW_INIT[5] register and is enabled with DESKEW_INIT[4].

The following lists the register addresses for each controller:

- Controller 0: 0x403 and 0x404
- Controller 1: 0x443 and 0x444
- Controller 2: 0x483 and 0x484
- Controller 3: 0x4C3 and 0x4C4

Periodic deskew is configured with the DESKEW_PER registers. The period of the Tx deskew calibration can be defined within the range of 1 to 128 frames. Periodic deskew can be enabled on either the rising or falling edge of VSYNC.

DESKEW_INIT[2:0] determines initial deskew width, i.e. Tskewcal in the initial skew-calibration mode. We recommend set DESKEW_INIT[2:0] =1 to guarantee it greater than minimum allowed value. DESKEW_PER[2:0] determines periodic deskew width, i.e. Tskewcal in the periodic skew-calibration mode. We recommend set DESKEW_PER[2:0] =1 to guarantee it greater than the minimum allowed value.

Note: Periodic deskew can be independently configured for any virtual channel in an enabled controller using the SKEW_PER_SEL[7:0] bitfield(s).

Table 265. D-PHY Deserializers Deskew Calibration Configuration Registers

Register	Bitfield	Bits	Description
MIPI_TX3	DESKEW_INIT[7:0]	7:0	**Initial deskew pattern settings** Bit 7: Auto initial deskew on/off Bit 6: Reserved Bit 5: when bit 4 = 1, any change of this bit will trigger one-time immediate initial skew. Bit 4: Manual initial on/off Bit 3: Reserved Bits [2:0]: Select initial deskew width: 1, 2, 3, … 8 * (32K) UI
MIPI_TX4	DESKEW_PER[7:0]	7:0	**Periodic deskew pattern settings** Bit 7: Period deskew calibration on/off Bit 6: Select generation on rising or falling edge of VS Bit [5:3]: Select periodic interval at every: 1, 2, 4, 8, … 128 frames Bit [2:0]: Select periodic deskew width: 1, 2, 3, … 8 * (1K) UI
MIPI_TX50	SKEW_PER_SEL[7:0]	7:0	**Periodic deskew select register**

ADI Confidential for new vision under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

			Bit 7: Select periodic deskew calibration for one or all virtual channels. • 0: Generate periodic deskew on all VC. • 1: on selected VC by bit 4:0 Bits [4:0] Virtual channel to generate periodic deskew calibration when Bit[7]=1

45.8 Debug Techniques

Note that external MIPI receivers use the calibration patterns generated by the GMSL2 D-PHY deserializers for calibration purposes. MIPI deskew debugging must consider all parts of the MIPI system.

45.8.1 GMSL2 D-PHY Serializers

If deskew status registers indicate that the deskew pattern is not received, ensure that the initial deskew pattern is sent before any data packets and that the deskew pattern transmitted by the MIPI source adheres to the MIPI D-PHY v1.4 specifications (i.e., deskew sync pattern and timing requirements).

If the MIPI receive indicates that the deskew calibration has failed, measure the skew between the clock and data lanes. The calibration circuit cannot guarantee skew compensation if the skew is greater than ±0.4UI. Check the MIPI rate to ensure that it is within the operating range of the deskew function (between 1.5Gbps and 2.5Gbps). Verify that the MIPI clock is stable and that any variation meets the MIPI specification. Verify that the initial deskew configuration procedure was followed and that an initial deskew pattern is transmitted before MIPI packets are sent.

45.8.2 GMSL2 D-PHY Deserializers

Repeat the process described above to continue debugging the MIPI system. Consult relevant documentation for the external MIPI receiver for further system debugging.

45.9 Feature Availability by Device Family

MIPI deskew is available on GMSL2 devices with MIPI D-PHY v1.2 ports (Table 266).

Table 266. MIPI D-PHY Deskew Support by Device Family

Device Family	MIPI Deskew
HDMI Serializers	
Advanced HDMI Serializers	
DSI Serializers	X
Advanced DSI Serializers	X
CSI-2 Serializers	X
Advanced CSI-2 Serializers	X
CSI-2 Camera Deserializers	X
CSI-2 Display Deserializers	X

Maxim Confidential
API Confidential for New Vision under NDA
Provided to GMSL2 Integrated Company Confidential Customers Under NDA

OLDI Deserializers	
eDP/DP Deserializers	
CSI-2 Quad Deserializers	X

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential, Provided to GMSL2_Customers Under NDA

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA

Appendix

46 Revision History

Revision Number	Date	Author	Changes
Version 00	24-Mar-17	N. Chimento	Initial Document
Version 01	24-Mar-17	N. Chimento	Added main Table of Contents
Version 02	7-Jul-17	N. Chimento	Added sections and began adding collateral
Version 03	18-Oct-17	N. Chimento	Continued to add information
Version 04	24-Oct-17	N. Chimento	Continued to add information and reorganize
Version 05	10-Nov-17	N. Chimento	Continued to add information and reorganize
Version 06	18-Jan-18	N. Chimento	Continued to add information and reorganize
Version 07	25-Mar-18	N. Chimento	Continued to add information and significant reorganization
Version 08	26-Mar-18	N. Chimento	Added in link integrity algorithms, and other appnotes
Version 09	4-May-18	N. Chimento	Added in various updates
Version 10	7-May-18	N. Chimento	Added in Quad Des programming guide
Version 11	15-Aug-18	N. Chimento	Added in sections and cleaned up existing sections.
Version 12	7-Oct-18	N. Chimento	Feedback from field and more editing. Additional sections added. Many sections reviewed
Version 13	1-Nov-18	N. Chimento	Additional sections added/edited
Version 14	18-Nov-18	N. Chimento	GMSL1 backwards compatibility, HDMI serializer video detection, and SPI added. Pattern generator and eDP Users guide updated
Version 15	26-Nov-18	N. Chimento	Minor edits for first customer release
Version 16	28-Nov-18	N. Chimento	Edit of mistake in I2C broadcasting section
Version 17	18-Mar-19	N. Chimento	Updated the following sections: Watermarking Asymmetric DSI Serializers Color Lookup Table (LUT)

ADI Confidential for New Revision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_ Customers Under NDA

			Audio – I2S and TDM GMSL2 protocol eDP Deserializer PCLK detection Splitter Mode Dualview Line Fault GMSL2 Error Pin Link Quality Procedure SPI Video Crossbar Spread spectrum PRBS Testing
Version 18	18-Jun-19	N. Chimento	Added Touch Applications section Added ADC Section Added GPIO rise/fall time section Added steps to debug RGMII/RMII Added Asymmetric DSI section Update to Link Lock section Update to Spread Spectrum section Update to Audio section Update to SPI section Update to FEC section Update to Link Integrity Tools section Update to FEC section Update to HDCP section Update to DSI section Update to Splitter Mode section Update to Link Protocol section Deleted Clock Options section – information now included in datasheets Various typos throughout were corrected
Version 19	9-July-19	N. Chimento	Added Mandatory Register Settings section Update to I2S Audio section Correction to GPIO programming example
Version 20	16-Jan-20	N. Chimento	Update to OLDI pixel swap mandatory register write section Update to Power up Section Update to Link Basics section – now “Introduction” Update to Spread Spectrum section Update to CRC section Update to ARQ section Update to watermarking section Minor revisions throughout
Version 21	25-Apr-20	N. Chimento	Major updates to the following sections: GMSL2 Overview GMSL2 Link Configurations GMSL2 Link Rate Configuration Device Power-UP Adaptive Equalization Resets

ADI Confidential for New Version under NDA
Analog Devices Confidential, Subject to Company Confidentiality Obligations
Maxim Integrated Company Confidential, Customers Under NDA
Provided to GMSL2_Customers Under NDA

			Spread Spectrum CRC and ARQ Line Fault Voltage Monitoring Clock and Frequency Reference Dualview Display Stream Compression (DSC) Forward Error Correction (FEC) Watermarking Color Look Up Table Sync Pulse GPIOs MFP rate adjustment Video Basics DSI Serializers eDP Deserializers Quad Deserializers Added Error Generator (ERRG) section Added EMI/EMC section
Version 22	19-Jan-21	N. Chimento G Broussard	Interim release of some section updates. All changes overridden by update v23.
Version 23	11-Aug-21	N. Chimento G. Broussard	Significant updates to the following sections: GMSL2 Link Basics Spread-Spectrum Clocking (SSC) Power Manager and Sleep Mode Clocks and Frequency Reference PRBS Testing Video Basics Camera Basics Video Pipes Dual-View Display Stream Compression (DSC) Forward Error Correction (FEC) HDCP Watermarking Video Timing Generator (VTG) and Video Pattern Generator (VPG) Video Crossbar Color Lookup Table (LUT) Frame Sync Video Sync Pulse Outputs Audio I2C/UART Touch-Screen Applications General-Purpose Input and Output (GPIO) Serial Peripheral Interface (SPI) RMII/RGMII (Ethernet) Bandwidth Calculations GMSL2 Link System Bandwidth GMSL2 Error Reporting (ERRB Pin)

ADI Confidential for Newvision under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

			CRC Error Detection and ARQ Error Correction Voltage Monitoring Line Fault Error Generator (ERRG) HDMI Serializers DSI Serializers Dual OLDI Deserializers eDP/DP Deserializers CSI-2 Serializers and Deserializers CSI-2 Quad Deserializers MIPI D-PHY Deskew
Version 24	8-Feb-22	N. Chimento	Major updates to following sections: DSI Dualview operation considerations Dual Link mode detail added Video Bandwidth calculations GPIO Priority setting removed GPIO broadcast configuration Spread spectrum configuration tables UART Bypass mode GMSL Link Lock eDP/DP Deserializers VTRG section Minor updates to following sections: CSI Deserializers – small improvements and new subsection – Video PRBS testing added MIPI D-PHY Deskew PRBS Testing
Version 25	4-May-22	N. Chimento	Major updates to following sections: Lock time optimization Minor updates to following sections: Watermarking – update to only support RGB888 datatype HDCP section Updated formatting from Maxim – Analog Devices Added section numbers
Version 26	5-May-22	N. Chimento	Major update to eDP/DP Deserializer section Minor updates to formatting and cross references

ADI Confidential for New Version under NDA
Analog Devices Confidential, Limited Distribution
Maxim Integrated Company Confidential, Limited Distribution
Provided to GMSL2_Customers Under NDA

47 Figures

FIGURE 1. GMSL2 SINGLE LINK MODE 6
FIGURE 2. GMSL2 SPLITTER MODE 7
FIGURE 3. REVERSE SPLITTER MODE 7
FIGURE 4. DUAL LINK MODE 8
FIGURE 5. OPTIMAL INITIALIZATION FOR LINK LOCK TIME 18
FIGURE 6. POC LINK LOCK OPTIMIZATION 21
FIGURE 7. RESET_LINK TIMING OPTIMIZATION 22
FIGURE 8. RELOCK OPTIMIZATION 22
FIGURE 9. PERIODIC ADAPT NOT ENABLED 25
FIGURE 10. PERIODIC ADAPT ENABLED 25
FIGURE 11. EXAMPLE SSC CENTER FREQUENCY SPREAD 30
FIGURE 12. SIMPLIFIED DIAGRAM OF FREQUENCY REFERENCE SYSTEM 37
FIGURE 13. RCLKOUT/DPLL ARCHITECTURE 37
FIGURE 14. EXAMPLE CRYSTAL LAYOUT 39
FIGURE 15. CRYSTAL REFERENCE 49
FIGURE 16. EXTERNAL CLOCK SOURCE 49
FIGURE 17. USING RCLKOUT TO SHARE A COMMON CRYSTAL 50
FIGURE 18. GMSL2 SERIALIZER VIDEO AND AUDIO PRBS GENERATOR/CHECKER BLOCKS 51
FIGURE 19. GMSL2 DESERIALIZER VIDEO AND AUDIO PRBS GENERATOR/CHECKER BLOCKS 52
FIGURE 20. VIDEO FRAME REGIONS 61
FIGURE 21. SYNC PULSES: DATA ENABLE AND THE VERTICAL AND HORIZONTAL SYNC SIGNALS 62
FIGURE 22. HORIZONTAL SYNC AND DATA ENABLE SIGNALS 62
FIGURE 23. VIDEO PIPES IN A GMSL2 SYSTEM 69
FIGURE 24. GMSL2 SERIALIZER VIDEO PIPE STRUCTURE 71
FIGURE 25. GMSL2 DESERIALIZER VIDEO PIPE STRUCTURE 71
FIGURE 26. EXAMPLE OF SIDE-BY-SIDE SYMMETRIC DUAL-VIEW 75
FIGURE 27. SINGLE FRAME TIMING 76
FIGURE 28. PIXEL-INTERLEAVED DUAL-VIEW MODE 77
FIGURE 29. SIDE-BY-SIDE DUAL-VIEW MODE 78
FIGURE 30. LINE-INTERLEAVED DUAL-VIEW MODE 79
FIGURE 31. ASYMMETRIC DUAL-VIEW 80
FIGURE 32. ASYMMETRIC DUAL-VIEW EXAMPLE 81
FIGURE 33. PREDEFINED RATIOS AND LINE PLACEMENT FOR SUPERFRAME PADDING 82
FIGURE 34. VIDEO TIMING ALONG THE VIDEO PATH THROUGH THE SERIALIZER 85
FIGURE 35. UNCOMPRESSED VIDEO FRAME WITH 91,480 RANDOM BIT ERRORS 103
FIGURE 36. COMPRESSED VIDEO FRAME WITH 1 BIT ERROR 103
FIGURE 37. ARCHITECTURE OF FEC IMPLEMENTATION IN GMSL DEVICES 104
FIGURE 38. BLOCK DIAGRAM OF A SINGLE MICROCONTROLLER SYSTEM 106
FIGURE 39. BLOCK DIAGRAM OF A DUAL MICROCONTROLLER SYSTEM 107
FIGURE 40. SURROUND VIEW SYSTEM WATERMARK APPLICATION 114
FIGURE 41. WATERMARK STATUS OUTPUTS 117
FIGURE 42. VIDEO PATH THROUGH THE VTG AND VPG IN GMSL2 SERIALIZERS 123
FIGURE 43. VTG TIMING PARAMETERS 128
FIGURE 44. VIDEO TIMING DEFINITION 130
FIGURE 45. HS AND DE SIGNAL COMPONENTS 130
FIGURE 46. VPG – GRADIENT PATTERN 133
FIGURE 47. VPG – CHECKERBOARD PATTERN 133
FIGURE 48. VIDEO CROSSBAR BLOCK DIAGRAM (SINGLE OUTPUT BIT) 138
FIGURE 49. RGB PIXEL DATA TRANSLATION USING COLOR LOOKUP TABLES (LUT) 145

Maxim Confidential
For Maxim Customers Only
Maxim Integrated Confidential
ADI Confidential for New Version under NDA
Analog Devices Confidential
Company Confidential
For ADI Customers Under NDA

FIGURE 50. VS SIGNAL PATH (FRAME SYNC NOT ENABLED) 149
FIGURE 51. VS SIGNAL PATH (FRAME SYNC ENABLED) 150
FIGURE 52. OVERLAP WINDOW (FSYNC AND VS SIGNALS) 153
FIGURE 53. TYPICAL AUDIO SYSTEM BLOCK DIAGRAM 166
FIGURE 54. TYPICAL GMSL2 AUDIO SYSTEM BLOCK DIAGRAM 166
FIGURE 55. TWO-CHANNEL STEREO 168
FIGURE 56. 8-CHANNEL TDM 168
FIGURE 57. DEFAULT INPUT AND OUTPUT AUDIO POLARITY 169
FIGURE 58. FORWARD AUDIO CHANNEL I2S BLOCK DIAGRAM 173
FIGURE 59. REVERSE CHANNEL I2S BLOCK DIAGRAM 174
FIGURE 60. FORWARD AND REVERSE CHANNEL I2S BLOCK DIAGRAM 175
FIGURE 61. TWO FORWARD CHANNEL I2S IN SPLITTER MODE BLOCK DIAGRAM 176
FIGURE 62. TWO REVERSE CHANNEL I2S IN SPLITTER MODE BLOCK DIAGRAM 179
FIGURE 63. FORWARD AUDIO MASTER SET: AUD_SINK_SRC (DESERIALIZER) AND AUD_RX_SINK_SRC (SERIALIZER) 181
FIGURE 64. REVERSE AUDIO MASTER SET: AUD_SINK_SRC (SERIALIZER) AND AUD_RX_SINK_SRC (DESERIALIZER) 182
FIGURE 65. TYPICAL GMSL2 SERIAL LINK SYSTEM WITH MAIN CONTROL CHANNEL 188
FIGURE 66. GMSL2 SYSTEM WITH MULTI-MASTER MAIN CONTROL CHANNEL (I2C) 189
FIGURE 67. I2C WRITE DATA TRANSFER FORMAT 193
FIGURE 68. I2C READ DATA TRANSFER FORMAT 193
FIGURE 69. I2C CLOCK STRETCHING 194
FIGURE 70. I2C ADDRESS TRANSLATION 200
FIGURE 71. I2C BROADCASTING 203
FIGURE 72. UART FRAME FORMAT 212
FIGURE 73. UART SYNCHRONIZATION FRAME 213
FIGURE 74. UART WRITE PROTOCOL FORMAT 213
FIGURE 75. UART READ PROTOCOL FORMAT 213
FIGURE 76. UART ACKNOWLEDGE FRAME FORMAT 214
FIGURE 77. PASS-THROUGH I2C/UART 218
FIGURE 78.TYPICAL TOUCH-SCREEN SYSTEM 227
FIGURE 79. TOUCH-SCREEN SYSTEM WITH MULTIPLE TOUCH-SCREEN INTERFACES 227
FIGURE 80. RECOMMENDED SETUP FOR INITIAL TOUCH-SCREEN SYSTEM VERIFICATION 228
FIGURE 81. TYPICAL GMSL2 TOUCH-SCREEN APPLICATION 228
FIGURE 82. SINGLE TOUCH SCREEN SETUP 231
FIGURE 83. SINGLE TOUCH SCREEN EXAMPLE SIGNAL ANALYSIS 232
FIGURE 84. SINGLE TOUCH SCREEN EXAMPLE TIMING ANALYSIS 233
FIGURE 85. MULTIPLE TOUCH SCREEN SETUP 234
FIGURE 86. MULTIPLE TOUCH SCREEN EXAMPLE SIGNAL ANALYSIS 236
FIGURE 87. MULTIPLE TOUCH SCREEN EXAMPLE TIMING ANALYSIS 237
FIGURE 88. BLOCK DIAGRAM OF GPIO TX AND RX BLOCKS MAPPING 242
FIGURE 89. GPIO FORWARDING – STANDARD MODE 243
FIGURE 90. GPIO FORWARDING – DELAY COMPENSATED MODE 243
FIGURE 91. EXAMPLE OF GPIO FORWARDING WITH THREE GPIOS 244
FIGURE 92. EXAMPLE OF A GPIO INPUT ROUTED TO MULTIPLE OUTPUTS 246
FIGURE 93. GMSL2 SPI INTERFACE 253
FIGURE 94. GMSL2 SPI IMPLEMENTATION 258
FIGURE 95. SPI NETWORK: TWO DESERIALIZERS AND ONE SERIALIZER 260
FIGURE 96. SPI NETWORK: TWO SERIALIZERS AND ONE DESERIALIZER 263
FIGURE 97. TYPICAL SPI APPLICATION 265
FIGURE 98. VIDEO AND ETHERNET PACKETS TRANSMITTED IN A GMSL2 LINK SYSTEM 267
FIGURE 99. RELATIONSHIP BETWEEN ETHERNET THROUGHPUT AND THE VIDEO PIXEL CLOCK 269
FIGURE 100. GMSL2 DEVICE CONNECTIONS TO ETHERNET PHY 270
FIGURE 101. GMSL2 DEVICE CONNECTIONS TO MAC/SOC 270
FIGURE 102. RMII TIMINGS WITH EDGE-ALIGNED DATA WHEN ADD_RX_DELAY = 0 AND ADD_TX_DELAY = 0 271

Maxim Confidential for New Version under NDA
Maxim Confidential Distribution Only
Subject to Change / Confidential
Provided to GMSL2 Customers Under NDA

FIGURE 103. RMII TIMINGS WITH EDGE-ALIGNED DATA WHEN ADD_RX_DELAY = 1 AND ADD_TX_DELAY = 1 271
FIGURE 104. TXC_DELAY USAGE 272
FIGURE 105. RMII HARDWARE CONFIGURATION 1 (PHY) 273
FIGURE 106. RMII HARDWARE CONFIGURATION 2 (PHY) 273
FIGURE 107. RMII HARDWARE CONFIGURATION 3 (MAC/SOC) 274
FIGURE 108. RMII BLOCK DIAGRAM 275
FIGURE 109. RMII CLOCK CIRCUIT BLOCK DIAGRAM 276
FIGURE 110. SENDING ETHERNET DATA ACROSS THE GMSL2 LINK 278
FIGURE 111. RECEIVING ETHERNET DATA ACROSS THE GMSL2 LINK 280
FIGURE 112. RGMII BLOCK DIAGRAM 284
FIGURE 113. RGMII HARDWARE CONFIGURATION (SOC) 285
FIGURE 114. CLOCK-TO-DATA SKEW 287
FIGURE 115. RXC_DELAY 287
FIGURE 116. TXC_DELAY 288
FIGURE 117. MDIO 290
FIGURE 118. LOCAL LOOPBACK 291
FIGURE 119. REMOTE LOOPBACK 292
FIGURE 120. ETHERSAM (Y.1564) TESTING 292
FIGURE 121. ETHERSAM SERVICE CONFIGURATION AND SERVICE PERFORMANCE TESTS RESULTS 293
FIGURE 122. ETHERSAM SUB-TESTS RESULTS 293
FIGURE 123. ETHERSAM SERVICES SUMMARY 294
FIGURE 124. GMSL2 SYSTEM BLOCK DIAGRAM 296
FIGURE 125. A GMSL2 SERIALIZER WITH TWO INDEPENDENT MIPI INPUTS 315
FIGURE 126. MIPI LANE CONFIGURATIONS 323
FIGURE 127. ERRB LOGIC 336
FIGURE 128. INTERNAL ERROR REPORTING MECHANISM 336
FIGURE 129. ERRB TRANSITION SYSTEM REACTION PROCESS 337
FIGURE 130. OPERATION OF GO-BACK-N ARQ WITH GO-BACK-3 367
FIGURE 131. ARQ PATH 368
FIGURE 132. VOLTAGE MONITOR AND INTERNAL POWER PINS 373
FIGURE 133. VOLTAGE MONITORING ERRB CONFIGURATION 376
FIGURE 134. LINE-FAULT CONFIGURATION 1: LOCAL-SIDE SERIALIZER (COAX MODE) 380
FIGURE 135. LINE-FAULT CONFIGURATION 2: LOCAL-SIDE DESERIALIZER (COAX MODE) 380
FIGURE 136. LINE-FAULT DETECTION CONFIGURATION: LOCAL-SIDE SERIALIZER (TWISTED-PAIR MODE) 381
FIGURE 137. BLOCK DIAGRAM OF THE VIDEO PATH THROUGH GMSL2 HDMI SERIALIZERS 388
FIGURE 138. EMBEDDED HMDI APPLICATION 389
FIGURE 139. CONSUMER PORT APPLICATION 389
FIGURE 140. EDID MEMORY IN THE SERIALIZER HDMI RX BLOCK 390
FIGURE 141. HDMI FAULT/ERROR REGISTER ARCHITECTURE 400
FIGURE 142. SIMPLIFIED BLOCK DIAGRAM OF DUAL PORT DSI SERIALIZER 403
FIGURE 143. GMSL2 DSI SERIALIZER – 2(1X4) MODE 404
FIGURE 144. GMSL2 DSI SERIALIZER – 1(1X4) MODE (MAX9283/85) 404
FIGURE 145. SIMPLIFIED BLOCK DIAGRAM OF DSI RX CONTROLLER 405
FIGURE 146. EXAMPLE OF HS BURST PACKETS WITH EOTP 406
FIGURE 147. DSI LONG PACKET STRUCTURE 406
FIGURE 148. DSI SHORT PACKET STRUCTURE 407
FIGURE 149. COLOR FORMAT OF RGB888 408
FIGURE 150. NON-BURST MODE TRANSMISSION WITH SYNC START AND SYNC END (SYNC PULSES) 410
FIGURE 151. NON-BURST MODE TRANSMISSION WITH SYNC START EVENTS (SYNC EVENTS) 411
FIGURE 152. DSI CONFIGURATION FLOWCHART 414
FIGURE 153. SINGLE DISPLAY APPLICATION 416
FIGURE 154. DUAL DSI INDEPENDENT DISPLAYS APPLICATION 416
FIGURE 155. DUPLICATED DISPLAY APPLICATION 417

Maxim Integrated Confidential
Subject to Change
Not for New Design
Confidential
Under NDA
Customers
MAX Confidential for New Design under NDA

FIGURE 156. SYMMETRIC DUAL-VIEW SPLITTER APPLICATION 418
FIGURE 157. ASYMMETRIC DUAL-VIEW SPLITTER APPLICATION 419
FIGURE 158. DUAL PORT "GANGED MODE" 420
FIGURE 159. 2X4 INTERNAL MAPPING DIAGRAM 428
FIGURE 160. 1X4 INTERNAL MAPPING DIAGRAM (MAX9283/85) 428
FIGURE 161. MIPI DPI DESKEW CONVERSION 432
FIGURE 162. DUAL-VIEW BLOCK VIDEO SPLITTING 435
FIGURE 163. LVDS PULSE POSITIONS 452
FIGURE 164. SINGLE OLDI PORT MAPPING (OLDI/FORMAT 1) 453
FIGURE 165. SINGLE OLDI PORT MAPPING (VESA/FORMAT 2) 453
FIGURE 166. DUAL OLDI PORT MAPPING (OLDI/FORMAT 1) 454
FIGURE 167. DUAL OLDI PORT MAPPING (VESA/FORMAT 2) 454
FIGURE 168. SERIALIZER SINGLE LINK MODE WITH OLDI DESERIALIZER SINGLE OR DUAL MODE 456
FIGURE 169. SERIALIZER DUAL-VIEW MODE WITH OLDI DESERIALIZER SPLITTER MODE 457
FIGURE 170. SINGLE VIDEO STREAM AND GMSL2 DUAL LINK MODE 457
FIGURE 171. SINGLE VIDEO STREAM WITH SERIALIZER REPLICATION 458
FIGURE 172. SINGLE VIDEO STREAM WITH OLDI DESERIALIZER REPLICATION 458
FIGURE 173. EDP/DP BLOCK DIAGRAM 488
FIGURE 174. VIDEO PATH IN GMSL2 EDP/DP DESERIALIZERS 499
FIGURE 175. VTRG BLOCK DIAGRAM 504
FIGURE 176. EDP/DP DESERIALIZER VIDEO PATH INCLUDING PCLK RATE CONTROL 505
FIGURE 177. EXAMPLE GMSL2 CSI-2 SYSTEM BLOCK DIAGRAM 527
FIGURE 178. MAX9295D/E CSI-2 SERIALIZER BLOCK DIAGRAM 529
FIGURE 179. MIPI CSI-2 CONTROLLER BLOCK DIAGRAM 530
FIGURE 180. MIPI CSI-2 PACKETS TO VIDEO CONVERSION (TWO FRAMES) 531
FIGURE 181. MAX9296B CSI-2 DESERIALIZER BLOCK DIAGRAM 533
FIGURE 182. MAX9295A SERIALIZER D-PHY RX – 1(1X4) MODE 538
FIGURE 183. MAX9295B SERIALIZER D-PHY RX – 1(1X4) MODE 538
FIGURE 184. MAX9295B SERIALIZER D-PHY RX – 2(1X2) MODE 539
FIGURE 185. MAX9295D/E SERIALIZER D-PHY RX – 2(1X4) MODE 540
FIGURE 186. MAX9295A SERIALIZER – LANE SWAP 543
FIGURE 187. GMSL2 CSI-2 SERIALIZER CONTROLLER-TO-PIPE ROUTING 545
FIGURE 188. GMSL2 CSI-2 VIDEO PIPES – SYSTEM ROUTING 550
FIGURE 189. CSI-2 VS ROUTING USE CASES 553
FIGURE 190. CASE 1.B) SERIALIZER – EMBEDDED DATA INPUT 554
FIGURE 191. CASE 1.B) DESERIALIZER – EMBEDDED DATA INPUT 554
FIGURE 192. CASE 2.A) SERIALIZER – FRAME-SYNCHRONIZED LINE-INTERLEAVED INPUT 556
FIGURE 193. CASE 2.A) DESERIALIZER – FRAME-SYNCHRONIZED LINE-INTERLEAVED INPUT 557
FIGURE 194. CASE 2.B) SERIALIZER – UNSYNCHRONIZED LINE-INTERLEAVED INPUT 558
FIGURE 195. CASE 2.B) DESERIALIZER – UNSYNCHRONIZED LINE-INTERLEAVED INPUT 558
FIGURE 196. CASE 2.C) SERIALIZER – FRAME-INTERLEAVED INPUT 559
FIGURE 197. CASE 2.C) DESERIALIZER – FRAME-INTERLEAVED INPUT 560
FIGURE 198. MAX9296 PIPE TO CONTROLLER ROUTING 569
FIGURE 199. MAX9295A/MAX9296A USING 2.3MP IMAGE SENSOR (ADVANCED USE CASE) 573
FIGURE 200. MAX9296A/B DESERIALIZER D-PHY RX – 2(1X4) MODE 583
FIGURE 201. MAX9296A/B DESERIALIZER D-PHY RX – 4(1X2) MODE 584
FIGURE 202. MAX9296B DESERIALIZER D-PHY RX – 1X4A + 2(1X2) MODE 585
FIGURE 203. MAX9296B DESERIALIZER D-PHY RX – 1X4B + 2(1X2) MODE 586
FIGURE 204. MAX9296B DESERIALIZER – LANE SWAP 587
FIGURE 205. DATA REPLICATION USING THE PHY COPY FUNCTION 590
FIGURE 206. MAX9295A/MAX9296A USING 2.3MP IMAGE SENSOR (SIMPLE USE CASE) 595
FIGURE 207. MAX9295A/MAX9296A USING 2.3MP IMAGE SENSOR (ADVANCED USE CASE) 598
FIGURE 208. MAX96712 VIDEO PATH – 2(1X4) D-PHY MODE 619

Maxim Integrated Confidential
Subject to Change Without Notice
Maxim Integrated Confidential
Maxim Integrated Confidential
Maxim Integrated Confidential

FIGURE 209. MAX96712 VIDEO PATH – 4(1X2) D-PHY MODE 620
FIGURE 210.MAX96712 VIDEO PATH – 2(1X4) C-PHY MODE 621
FIGURE 211. MAX96712 VIDEO PATH – 4(1X2) C-PHY MODE....621
FIGURE 212. VIDEO PIPE SELECTION BLOCK DIAGRAM 625
FIGURE 213. VIDEO PIPE TO MIPI CONTROLLER MAPPING BLOCK DIAGRAM....629
FIGURE 214. D-PHY LANE SWAP EXAMPLE....640
FIGURE 215. D-PHY POLARITY SWAP EXAMPLE....641
FIGURE 216. C-PHY LANE SWAP EXAMPLE....641
FIGURE 217. C-PHY POLARITY SWAP EXAMPLE....641
FIGURE 218. BUILDING THE CLINK ON HIM-CAPABLE GMSL1 SERIALIZER 658
FIGURE 219. BUILDING THE CLINK ON NON-HIM-CAPABLE GMSL1 SERIALIZER 662
FIGURE 220. VIDEO PIPE INPUT AND CONCATENATED SUPERFRAME OUTPUT 668
FIGURE 221. CONCATENATION (4WXH AND WX4H)....669
FIGURE 222. SUPERFRAME AGGREGATION 670
FIGURE 223. CSI-2 QUAD DESERIALIZER I2C/UART ROUTING....687
FIGURE 224. QUAD DESERIALIZER I2C PORT ACCESS....688
FIGURE 225. CSI-2 QUAD DESERIALIZER I2C/UART CROSSOVER SWITCHING 690
FIGURE 226. I2C BROADCASTING (QUAD DESERIALIZER) 694
FIGURE 227. GMSL2 SPI INTERFACE....701
FIGURE 228. CONCATENATED VIDEO STREAM FROM THREE CAMERAS 711
FIGURE 229. CONCATENATED VIDEO STREAM WITH MASKED VIDEO....711
FIGURE 230. MIPI D-PHY DESKEW WITHIN A GMSL2 SYSTEM....722
FIGURE 231. HIGH-SPEED DATA TRANSMISSION IN NORMAL MODE....728
FIGURE 232. HIGH-SPEED DESKEW CALIBRATION....729
FIGURE 233. SKEW CALIBRATION (DETAILED VIEW)....729

ADI Confidential for New Revision under NDA
Analog Devices Confidential Distribution only under NDA
Maxim Integrated Company Confidential
Provided to GMSL2_Customers Under NDA

48 Tables

TABLE 1. GMSL2 SERIAL LINK RATES BY PART NUMBER 5
TABLE 2. GMSL2 LINK RATE CONFIGURATION 9
TABLE 3. MANUAL LINK MODE CONFIGURATION 9
TABLE 4. GMSL2 SERIAL LINK CONFIGURATION BY PART NUMBER 13
TABLE 5. LINK LOCK TIME GUIDANCE 21
TABLE 6. GMSL2 TX SSC CONFIGURATION FOR FIXED PPM 31
TABLE 7. SERIALIZER/DESERIALIZER OUTPUT SPREAD SPECTRUM REGISTER CONFIGURATION 31
TABLE 8. GMSL1 SPREAD PERCENTAGE LIMITATION 32
TABLE 9. GMSL1 LINK SPREAD SPECTRUM REGISTERS 33
TABLE 10. EXAMPLE DC CHARACTERISTICS OF X1/OSC AND X2 PINS 38
TABLE 11. EXTERNAL CLOCK DC CHARACTERISTICS 39
TABLE 12. EXTERNAL CLOCK AC CHARACTERISTICS 40
TABLE 13. EXAMPLE RCLKOUT CHARACTERISTICS (MAX9295D) 40
TABLE 14. CONFIGURATION REGISTERS FOR RCLKOUT 45
TABLE 15. CONFIGURATION REGISTERS FOR PREDEFINED DPLL FREQUENCIES 45
TABLE 16. CONFIGURATION REGISTERS FOR ARBITRARY DPLL FREQUENCIES 46
TABLE 17. ALTERNATE CONFIGURATION REGISTERS FOR RCLKOUT 46
TABLE 18. ALTERNATE CONFIGURATION REGISTERS FOR PREDEFINED DPLL FREQUENCIES 47
TABLE 19. ALTERNATE CONFIGURATION REGISTERS FOR ARBITRARY DPLL FREQUENCIES 48
TABLE 20. CLOCK REFERENCE OUTPUT SUPPORT BY DEVICE FAMILY 50
TABLE 21. VIDEO PRBS REGISTERS 53
TABLE 22. HDMI SERIALIZER TO OLDI DESERIALIZER VIDEO PRBS TEST SEQUENCE 54
TABLE 23. AUDIO PRBS REGISTERS 55
TABLE 24. OBSERVATION TIME REQUIRED FOR 95% CONFIDENCE INTERVAL OF DIFFERENT BER (@ 6GBPS) 56
TABLE 25. PRBS SUPPORT BY DEVICE FAMILY 57
TABLE 26. VIDEO PIPE FEATURE SUPPORT BY DEVICE FAMILY 74
TABLE 27. PIXEL-INTERLEAVED DUAL-VIEW CONFIGURATION REGISTERS 83
TABLE 28. SIDE-BY-SIDE DUAL-VIEW CONFIGURATION REGISTERS 84
TABLE 29. LINE-INTERLEAVED DUAL-VIEW CONFIGURATION REGISTERS 86
TABLE 30. DUAL-VIEW SUPPORT BY DEVICE FAMILY 90
TABLE 31. VIDEO TIMINGS FOR PRE-DEFINED VIDEO PROFILES 93
TABLE 32. VIDEO PROFILE: SERIALIZER CONFIGURATION 94
TABLE 33. VIDEO PROFILE: DESERIALIZER CONFIGURATION 94
TABLE 34. SERIALIZER CONFIGURATION FOR DSC ENCODER SELF-TEST (FOR VIDEO PROFILE RESOLUTIONS) 95
TABLE 35. SERIALIZER CONFIGURATION FOR DSC ENCODER SELF-TEST (FOR NON-VIDEO PROFILE RESOLUTIONS) 96
TABLE 36. DSC ENCODER STATUS REGISTERS 98
TABLE 37. DSC ENCODER ERROR OUTPUT REGISTERS 98
TABLE 38. DSC SELF-TEST STATUS REGISTERS 99
TABLE 39. DSC ENCODER ERRB OUTPUT 99
TABLE 40. DSC DECODER STATUS REGISTERS 99
TABLE 41. DSC DECODER ERROR OUTPUT REGISTERS 101
TABLE 42. DSC DECODER ERRB OUTPUT 101
TABLE 43. DSC SUPPORT BY PART NUMBER 102
TABLE 44. FEC SUPPORT BY PART NUMBER 111
TABLE 45. WATERMARKING SUPPORT BY DEVICE FAMILY 122
TABLE 46. VTG OPERATION MODE CONFIGURATION REGISTERS 124
TABLE 47. ADDITIONAL CSI-2 SERIALIZER PCLK REGISTER SETTINGS 127
TABLE 48. VIDEO PATTERN PCLK SELECTION 127
TABLE 49. TIMING PARAMETER CONFIGURATION REGISTERS 129

Maxim Confidential for New Revision under NDA
Maxim Confidential – Do Not Distribute Outside Company Without Signed NDA
Provided to GMSL2 Customers Under NDA

TABLE 50. RECOMMENDED REGISTER SETTINGS FOR VTG PARAMETERS (STANDARD VIDEO TIMINGS) 131
TABLE 51. VPG CONFIGURATION REGISTERS FOR SERIALIZERS 134
TABLE 52. VPG CONFIGURATION REGISTERS FOR QUAD DESERIALIZERS 135
TABLE 53. VTG/VPG SUPPORT BY DEVICE FAMILY 136
TABLE 54. VIDEO CROSSBAR OUTPUT BITMAPPING REGISTERS 137
TABLE 55. RGB COLOR MAP 140
TABLE 56. RAW COLOR MAP 141
TABLE 57. YUV/YCRCB COLOR MAP 142
TABLE 58. UDP8/GENERIC8 COLOR MAP 143
TABLE 59. VIDEO CROSSBAR SUPPORT BY DEVICE FAMILY 144
TABLE 60. OLDI DESERIALIZER LUT REGISTERS 146
TABLE 61. EDP/DP DESERIALIZER LUT REGISTERS 147
TABLE 62. COLOR LOOKUP TABLE SUPPORT BY DEVICE FAMILY 148
TABLE 63. FRAME SYNC MODE (FSYNC_MODE) 151
TABLE 64. FRAME SYNC MODES AND METHODS (GMSL2) 152
TABLE 65. FRAME SYNC MODES AND METHODS (GMSL1) 152
TABLE 66. FSYNC METHODS OF OPERATION (FSYNC_METH) 153
TABLE 67. FRAME SYNC SUPPORT BY PART NUMBER 158
TABLE 68. HDMI SERIALIZERS AND ADVANCED HDMI SERIALIZERS 160
TABLE 69. DSI SERIALIZERS, ADVANCED DSI SERIALIZERS, AND ADVANCED CSI-2 SERIALIZERS 160
TABLE 70. CSI-2 SERIALIZERS 160
TABLE 71. CSI-2 CAMERA DESERIALIZERS (EXCEPT THE MAX96784) 161
TABLE 72. CSI-2 DISPLAY DESERIALIZERS AND THE MAX96784 161
TABLE 73. OLDI DESERIALIZERS 161
TABLE 74. EDP/DP DESERIALIZERS 162
TABLE 75. CSI-2 QUAD DESERIALIZERS 162
TABLE 76. SYNC PULSE OUTPUTS SUPPORT BY DEVICE FAMILY 164
TABLE 77. GMSL2 DEVICE AUDIO PINOUTS 167
TABLE 78. CONFIGURATION OF FORWARD CHANNEL AUDIO ON CHANNEL X 173
TABLE 79. CONFIGURATION OF FORWARD CHANNEL AUDIO ON CHANNEL Y 173
TABLE 80. CONFIGURATION OF REVERSE CHANNEL AUDIO 174
TABLE 81. FORWARD AND REVERSE CHANNEL AUDIO 175
TABLE 82. TWO FORWARD CHANNEL AUDIO IN SPLITTER MODE 176
TABLE 83. REVERSE CHANNEL AUDIO 179
TABLE 84. CONFIGURATION FOR FOUR-PIN BIDIRECTIONAL AUDIO – FORWARD AUDIO MASTER 181
TABLE 85. CONFIGURATION FOR FOUR-PIN BIDIRECTIONAL AUDIO – REVERSE AUDIO MASTER 182
TABLE 86. FORWARD CHANNEL AUDIO PRBS TEST – INTERNAL OSCILLATOR 184
TABLE 87. FORWARD CHANNEL AUDIO PRBS TEST – EXTERNAL AUDIO CLOCK 184
TABLE 88. REVERSE CHANNEL AUDIO PRBS TEST – INTERNAL OSCILLATOR 184
TABLE 89. REVERSE CHANNEL AUDIO PRBS TEST – EXTERNAL AUDIO CLOCK 184
TABLE 90. AUDIO SUPPORT BY DEVICE FAMILY 187
TABLE 91. I2C/UART SELECTION STATUS REGISTER 191
TABLE 92. REMOTE/LOCAL CONTROL CHANNEL AND DEVICE CONFIGURATION REGISTERS 192
TABLE 93. NUMBER OF I2C-TO-I2C LINKS CONFIGURATION REGISTERS 192
TABLE 94. I2C ACKNOWLEDGE BIT AND TIME-OUT STATUS (READ-ONLY REGISTERS) 193
TABLE 95. INTERNAL I2C SLAVE CONFIGURATION REGISTERS 195
TABLE 96. INTERNAL I2C MASTER CONFIGURATION REGISTERS 195
TABLE 97. I2C ADDRESS CONFIGURATION 196
TABLE 98. DEFAULT VALUES OF TR3 REGISTERS FOR MAX96754 197
TABLE 99. DEFAULT VALUES OF TR3 REGISTERS FOR MAX9295E 198
TABLE 100. I2C ADDRESS TRANSLATOR CONFIGURATION REGISTERS 199
TABLE 101. UART INITIAL OUTPUT DELAY CONFIGURATION REGISTER 207
TABLE 102. UART BYPASS MODE CONFIGURATION REGISTERS 208

Maxim Confidential For New Revision Under NDA
Maxim Integrated Confidential Property Of Maxim Integrated Company Confidential Under NDA Customers

TABLE 103. GPIO ID FOR MS FUNCTIONALITY 210
TABLE 104. UART RX SOURCE ARBITRATION TIME-OUT CONFIGURATION REGISTER 211
TABLE 105. UART DETECTED BIT LENGTH (READ-ONLY REGISTERS) 215
TABLE 106. UART RX/TX FIFO OVERFLOW STATUS (READ-ONLY REGISTERS) 215
TABLE 107. I2C/UART CONTROL CHANNEL SUPPORT BY DEVICE FAMILY 216
TABLE 108. PASS-THROUGH I2C CHANNELS ENABLE REGISTERS 219
TABLE 109. PASS-THROUGH I2C INTERNAL SLAVE CONFIGURATION REGISTERS 219
TABLE 110. PASS-THROUGH I2C INTERNAL MASTER CONFIGURATION REGISTERS 220
TABLE 111. PASS-THROUGH I2C ACKNOWLEDGE AND TIME-OUT STATUS (READ-ONLY REGISTERS) 220
TABLE 112. PASS-THROUGH I2C/UART DEVICE PIN ASSIGNMENTS AND CHANNEL CONNECTIONS CONFIGURATION REGISTERS 221
TABLE 113. PASS-THROUGH I2C ADDRESS TRANSLATOR CONFIGURATION REGISTERS 222
TABLE 114. PASS-THROUGH UART CHANNELS ENABLE REGISTERS 223
TABLE 115. PASS-THROUGH UART PARITY CHECK CONFIGURATION REGISTERS 223
TABLE 116. PASS-THROUGH UART BIT RATE CONFIGURATION REGISTERS 224
TABLE 117. PASS-THROUGH UART RX/TX FIFO OVERFLOW STATUS (READ-ONLY REGISTERS) 224
TABLE 118. PASS-THROUGH I2C/UART CHANNEL SUPPORT BY DEVICE FAMILY 226
TABLE 119. MFP SPEED CONTROL REGISTERS 249
TABLE 120. TYPICAL RISE/FALL TIMES FOR GMSL2 DEVICES 249
TABLE 121. SPI MINIMUM TIMING REQUIREMENTS 257
TABLE 122. RMII/RGMII RATE, REF CLOCK FREQUENCY, AND TX/RX SIGNALS 268
TABLE 123. RGMII-TO-RMII PIN CROSS REFERENCE 268
TABLE 124. LAB TESTING RESULTS OF PIXEL CLOCK VS. RMII/RGMII RATE 269
TABLE 125. RMII/RGMII SPEED SETTING REGISTERS 275
TABLE 126. RMII/RGMII CLOCK CONFIGURATION REGISTERS 276
TABLE 127. RMII/RGMII ENABLE REGISTERS 276
TABLE 128. RMII/RGMII PACKET CONFIGURATION (TX) 278
TABLE 129. RMII/RGMII TX OVERFLOW STATUS REGISTER 279
TABLE 130. RMII/RGMII RX FRAME DELAY 280
TABLE 131. RMII/RGMII INTERFRAME GAP 281
TABLE 132. RMII READD PREAMBLE AT GMSL2 RX 281
TABLE 133. RMII/RGMII RX OVERFLOW AND UNDERFLOW STATUS REGISTERS 281
TABLE 134. RMII/RGMII CONFIGURATION REGISTERS FOR SERIALIZER 282
TABLE 135. RMII/RGMII CONFIGURATION REGISTERS FOR DESERIALIZER 282
TABLE 136. RMII/RGMII TX INTERFACE RISE TIME SETTINGS 283
TABLE 137. RISE TIME PROGRAMMING REGISTERS BY DEVICE FAMILY 283
TABLE 138. RGMII MODE ENABLE REGISTERS 286
TABLE 139. RGMII SPEED SETTINGS 286
TABLE 140. RGMII CLOCK-TO-DATA SKEW DELAY 287
TABLE 141. 1000MBPS RGMII CONFIGURATION REGISTERS FOR SERIALIZER 288
TABLE 142. 1000MBPS RGMII CONFIGURATION REGISTERS FOR DESERIALIZER 289
TABLE 143. MDIO WRITE EXAMPLE 291
TABLE 144. MDIO READ EXAMPLE 291
TABLE 145. RMII/RGMII SUPPORT BY PART NUMBER 294
TABLE 146. EXAMPLE CALCULATIONS FOR 1080P, RGB888 VIDEO FOR AN HDMI SERIALIZER AND AN EDP/DP DESERIALIZER 297
TABLE 147. MAXIMUM VIDEO PAYLOAD FOR GMSL2 MODES 298
TABLE 148. BANDWIDTH LIMITING FACTORS 300
TABLE 149. MAXIMUM CSI-2 LANE RATES (MBPS) IN CONSTANT BPP MODE 310
TABLE 150. MIPI D-PHY AND C-PHY LANE CONFIGURATIONS AND OUTPUT BANDWIDTH 323
TABLE 151. PCLK VALUE DETECTION 324
TABLE 152. DSC DECODER ERROR RELATED REGISTERS 355
TABLE 153. PMX ERROR RELATED REGISTERS 361
TABLE 154. 16-BIT CRC REGISTERS 369
TABLE 155. 16-BIT ARQ REGISTERS 370

Maxim Confidential
Restricted to Maxim Confidential Under NDA
Maxim Confidential for New Revision Digitally Signed
Maxim Confidential for New Revision under NDA

TABLE 156. CRC/ARQ AVAILABILITY BY DEVICE FAMILY....371
TABLE 157. POWER SUPPLY MONITORING FUNCTIONS AVAILABLE....372
TABLE 158. VOLTAGE MONITORING FEATURE AVAILABILITY BY DEVICE FAMILY....377
TABLE 159. LINE-FAULT SIGNAL ASSIGNMENT TO SIO, AND RESISTORS IN COAX MODE....380
TABLE 160. LINE FAULT PAIR #1 (LMN0/LMN1) RESISTOR VALUES AND CONNECTION CHOICES IN STP MODE....381
TABLE 161. LINE FAULT PAIR #2 (LMN2/LMN3) RESISTOR VALUES AND CONNECTION CHOICES IN STP MODE....381
TABLE 162. REGISTER MAPPING AND DESCRIPTIONS FOR THE LINE-FAULT REGISTERS....382
TABLE 163. LINE-FAULT DETECTION DECODE TABLE....382
TABLE 164. ERRG PARAMETERS (REGISTER TX2)....385
TABLE 165. ERRG ENABLE (REGISTER TX1)....385
TABLE 166. ERRG ENABLE (REGISTER GMSL__X:TX1)....385
TABLE 167. AVAILABILITY BY DEVICE FAMILY....386
TABLE 168. PIXEL REPETITION REMOVAL....396
TABLE 169. GMSL2 HDMI SERIALIZER DEBUG REGISTERS....397
TABLE 170. HDMI RX INTERRUPT STATUS REGISTERS....399
TABLE 171. GMSL2 HDMI RX INTERRUPT ENABLE....400
TABLE 172. GMSL2 HDMI SERIALIZER INTERRUPT ENABLE REGISTERS....401
TABLE 173. HDMI DEVICE FAMILY PART NUMBERS....401
TABLE 174. SUPPORTED MIPI DSI PACKETS....407
TABLE 175. DSI GMSL SUPPORTED APPLICATIONS....415
TABLE 176. MAXIMUM MIPI LANE RATES FOR 6GBPS LINK....421
TABLE 177. INTERNAL MIPI PHY AND CONTROLLER MAPPING....429
TABLE 178. DEFAULT 2X4 MAPPING....429
TABLE 179. DPI DESKEW REGISTERS FOR EACH MIPI PORT....433
TABLE 180. EXAMPLE TIMING PARAMETERS FOR DPI DESKEW....434
TABLE 181. VERTICAL BLANKING RATIO REGISTER....436
TABLE 182. APPLICATION EXAMPLE REGISTERS....442
TABLE 183. GMSL2 DSI SERIALIZERS FEATURE AVAILABILITY....449
TABLE 184. OLDI INTERFACE PIN DESCRIPTIONS....451
TABLE 185. OLDI PORT CONFIGURATION REGISTERS....459
TABLE 186. OLDI OUTPUT POLARITY INVERSION AND PIN MAPPING REGISTERS....460
TABLE 187. PRBS REGISTERS....461
TABLE 188. LUT REGISTERS....461
TABLE 189. GPIO SYNC SIGNAL OUTPUT....461
TABLE 190. DUAL OLDI DESERIALIZER DEVICE FAMILY PART NUMBERS....464
TABLE 191. EDP/DP DESERIALIZER REGISTER TABLE....495
TABLE 192. VTRG CONFIGURATION REGISTERS....501
TABLE 193. VTRG CONFIGURATION REGISTERS....507
TABLE 194. VTRG DEBUG REGISTERS....508
TABLE 195. HS AND VS STANDARD DEVIATIONS FOR TWO COMMON RESOLUTIONS....511
TABLE 196. VTRG VPG CONFIGURATION REGISTERS....511
TABLE 197. PROGRAMMING STEPS TO ENABLE VTRG....514
TABLE 198. LEGACY VTRG DEBUG REGISTERS....519
TABLE 199. DP LINK TRAINING STATUS AND ERROR REGISTER DETAILS....522
TABLE 200. EDP/DP CONTROLLER STATE MACHINE DEBUG REGISTERS....523
TABLE 201. EDP/DP DESERIALIZER DEVICE FAMILY PART NUMBERS....525
TABLE 202. GMSL2 CSI-2 SERIALIZER PORT COMPARISON....535
TABLE 203. CSI-2 SERIALIZERS: MIPI PORTS BY PART NUMBER....536
TABLE 204. GMSL2 CSI-2 SERIALIZER PHY CONFIGURATION REGISTERS....537
TABLE 205. GMSL2 CSI-2 SERIALIZER LANE MAP AND POLARITY INVERSION REGISTERS....541
TABLE 206. MAX9295A SERIALIZER LANE SWAP EXAMPLE REGISTER WRITES....543
TABLE 207. GMSL2 CSI-2 SERIALIZER CONTROLLER-TO-PIPE MAPPING REGISTERS....546
TABLE 208. TYPICAL CSI-2 DATA TYPE HEX CODES....548

ADI Confidential for New Revision under NDA
Maxim Integrated Confidential
Provided to GMSL2 Customers Under NDA

TABLE 209. GMSL2 CSI-2 SERIALIZER DOUBLE MODE REGISTERS 561
TABLE 210. GMSL2 CSI-2 SERIALIZER ZERO PADDING REGISTERS 562
TABLE 211. GMSL2 CSI-2 SERIALIZER HEARTBEAT DISABLE REGISTERS 563
TABLE 212. GMSL2 CSI-2 SERIALIZER PIPE ENABLE AND STREAM ID REGISTERS 564
TABLE 213. GMSL2 CSI-2 SERIALIZER ERROR REGISTERS 565
TABLE 214. GMSL2 CSI-2 SERIALIZER VIDEO PIPE AND CONTROLLER STATUS AND ERROR BITS 567
TABLE 215. GMSL2 CSI-2 DESERIALIZER VIDEO PIPE STREAM ID CONFIGURATION REGISTERS 569
TABLE 216. GMSL2 CSI-2 DESERIALIZER VS CROSSBAR REGISTERS 569
TABLE 217. GMSL2 CSI-2 DESERIALIZER BPP, DT, AND VC OVERRIDE REGISTERS 570
TABLE 218. GMSL2 CSI-2 DESERIALIZER PIPE-TO-CONTROLLER MAPPING 571
TABLE 219. GMSL2 CSI-2 DESERIALIZER FRAME-START/END OVERRIDE 576
TABLE 220. GMSL2 CSI-2 DESERIALIZER DOUBLE MODE REGISTERS 577
TABLE 221. EXAMPLE DOUBLE MODE SETTINGS FOR A SERIALIZER AND DESERIALIZER PAIR 578
TABLE 222. GMSL2 CSI-2 DESERIALIZER HEARTBEAT DISABLE REGISTERS (MAX9296 ONLY) 579
TABLE 223. CSI-2 DESERIALIZERS: MIPI PORTS BY PART NUMBER 580
TABLE 224. GMSL2 CSI-2 DESERIALIZER PORT CONFIGURATION REGISTERS 581
TABLE 225. MAX9296B DESERIALIZER LANE SWAP EXAMPLE REGISTER WRITES 588
TABLE 226. GMSL2 CSI-2 DESERIALIZER STATUS AND ERROR REGISTERS 591
TABLE 227. USE CASES SUPPORTED BY THE CSI-2 CONFIGURATION TOOL 594
TABLE 228. GMSL2 CSI-2 SERIALIZER DEFAULT ROUTING 594
TABLE 229. GMSL2 CSI-2 DESERIALIZER DEFAULT ROUTING 594
TABLE 230. CSI-2 SUPPORT BY DEVICE FAMILY 608
TABLE 231. GMSL2 CSI-2 QUAD DESERIALIZER LINK RATES 609
TABLE 232. LINK INITIALIZATION REGISTERS 623
TABLE 233. VIDEO PIPE SELECTION REGISTERS 626
TABLE 234. VIDEO PIPE TO MIPI PHY CONTROLLER REGISTERS 629
TABLE 235. EXTENDED VIRTUAL CHANNELS REGISTERS 632
TABLE 236. SOURCE MAPPING (EXTENDED VC) 634
TABLE 237. DESTINATION MAPPING (EXTENDED VC) 634
TABLE 238. MIPI PHY SETTINGS REGISTERS 636
TABLE 239. CONFIGURATION STEPS FOR PAIRING HIM-CAPABLE GMSL1 SERIALIZERS 659
TABLE 240. CONFIGURATION STEPS FOR PAIRING NON-HIM-CAPABLE GMSL1 SERIALIZERS 663
TABLE 241. SOFTWARE OVERRIDE REGISTERS 665
TABLE 242. MIPI PHY REPLICATION REGISTERS 667
TABLE 243. FRAME CONCATENATION REGISTERS 670
TABLE 244. FRAME SYNC REGISTERS 672
TABLE 245. VIDEO PATTERN REGISTERS 676
TABLE 246. PLCK SETTINGS REGISTERS 678
TABLE 247. VIDEO PATTERN PCLK SELECTION 679
TABLE 248. MIPI PRBS GENERATOR REGISTERS 680
TABLE 249. MIPI VIDEO PRBS GENERATOR AND CHECKER REGISTERS 681
TABLE 250. CONTROL CHANNEL REGISTERS 691
TABLE 251. I2C BROADCASTING (QUAD) EXAMPLE – SERIALIZER 696
TABLE 252. I2C BROADCASTING (QUAD) EXAMPLE – IMAGE SENSOR 696
TABLE 253. LINK INITIALIZATION REGISTERS USED FOR LINK SELECTION 698
TABLE 254. SPI SETTING REGISTERS 702
TABLE 255. MIPI D-PHY DESKEW REGISTERS 704
TABLE 256. VIDEO PIPE AND MIPI CONTROLLER RESET REGISTERS 705
TABLE 257. MIPI C-PHY TIMING SETTING REGISTERS 707
TABLE 258. SYNC PULSE OUTPUT REGISTERS 708
TABLE 259. SYNC SIGNAL STATUS REGISTERS 709
TABLE 260. VIDEO MASK OUTPUT REGISTERS 712
TABLE 261. ERROR-CORRECTING CODE (ECC) MEMORY CONFIGURATION REGISTERS 713

Maxim Integrated Confidential
Protected by GMSL2 Company Confidential
MAX Confidential for New Revision under NDA
Digital Under NDA

TABLE 262. MIPI PACKET COUNT REGISTERS 715
TABLE 263. CSI-2 QUAD DESERIALIZER DEVICE FAMILY PART NUMBERS 716
TABLE 264. D-PHY SERIALIZERS DESKEW CALIBRATION CONFIGURATION REGISTERS 731
TABLE 265. D-PHY DESERIALIZERS DESKEW CALIBRATION CONFIGURATION REGISTERS 732
TABLE 266. MIPI D-PHY DESKEW SUPPORT BY DEVICE FAMILY 733

ADI Confidential for Newvision under NDA
Analog Devices Confidential, Distribute only under NDA
Maxim Integrated Company Confidential,
Provided to GMSL2_Customers Under NDA