

## General Description

The MAX96717 GMSL™ serializer receives video on an MIPI CSI-2 interface and outputs it on a GMSL2 serial link transceiver. Simultaneously, it sends and receives bidirectional control channel data across the same GMSL2 link. GMSL2 data can be transported over Coaxial or Shielded-Twisted Pair (STP) cables. It operates at a fixed rate of 3Gbps or 6Gbps in the forward direction, and 187.5Mbps in the reverse direction. The device is programmed through a local I<sup>2</sup>C/UART interface or across the link from a matching deserializer. The MAX96717 includes two I<sup>2</sup>C/UART pass-through channels, flexible GPIO, SPI tunnel, a built-in ADC, temperature sensor, and an extensive set of diagnostics for functional safety. Operation is specified over the automotive temperature range of -40°C to +105°C and the device is AEC-Q100 Grade 2 qualified.

Data can be transmitted over low-cost 50Ω Coax or 100Ω STP cables that meet the GMSL2 channel specification. [Table 1](#) provides guidance to typical maximum lengths of commonly used automotive cables. Contact the factory for the GMSL2 channel specification.

Table 1. Typical Maximum Cable Length vs. Attenuation

	3.2mm Ø 50Ω Coax, Foam Dielectric	2.7mm Ø 50Ω Coax, Solid Dielectric	100Ω Shielded Twisted Pair, AWG26
Attenuation at 3GHz (Typ, Room Temp)	0.9dB/m	1.6dB/m	1.8dB/m
Attenuation at 3GHz (Max, Aged, 105°C)	1.1dB/m	2.0dB/m	2.2dB/m
GMSL Fwd/Rev Data Rate	Typical Maximum Cable Length at 105°C		
3Gbps/187.5Mbps	20m	10m	11m
6Gbps/187.5Mbps	15m	9m	8m

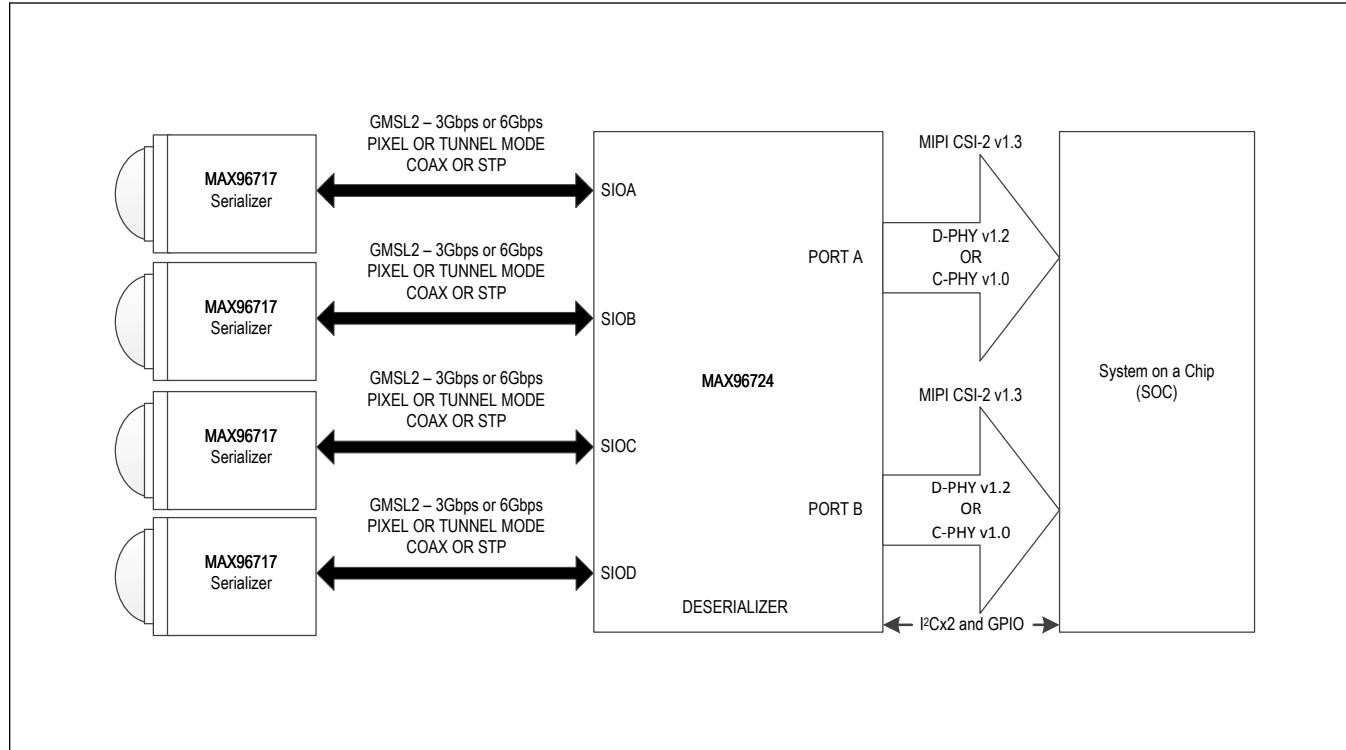
## Applications

- Advanced Driver Assistance Systems (ADAS)
- 8Mp 40fps Forward-Vision Camera (FVC) Systems
- Surround View Systems (SVS)
- Driver Monitor Systems (DMS)
- Rear-View Camera (RVC)
- Systems with Multiple Synchronized Cameras

## Benefits and Features

- Automotive Grade High Speed Link
  - 21.0dB at 3GHz (6Gbps) Max Insertion Loss
  - 19.5dB at 1.5GHz (3Gbps) Max Insertion Loss
  - Auto Adapt for Changes in Channel Conditions
  - Operates at -40°C to +105°C Ambient
- Four-Lane MIPI CSI-2 v1.3 Input Port
  - MIPI D-PHY v1.2 Receivers rated at 2.5Gbps per Lane
  - Supports any CSI-2 Data Type in Tunneling Mode
  - Cyclic Redundancy Check (CRC), and Error Correction Code (ECC)
  - 16 Virtual Channels
  - Polarity Flip and Data Lane Reassignment
- Full-Duplex Communication Over a Single Wire
- 3Gbps or 6Gbps Forward-Link Rates, and 187.5Mbps Reverse-Link Rate
- ASIL-B Compliant
  - End-to-End Data Integrity through CRC in Tunneling Mode
  - R-S Forward-Error Correction (FEC) for Protection of Forward Video and Control-Channel Data
  - CRC Protection of Side-Channel and Video Data
  - Power-on Self Test
- Concurrent Side Channel for Device Configuration and Communication with Peripherals
  - I<sup>2</sup>C/UART, Pass-Through I<sup>2</sup>C/UART, SPI, GPIO, and Register-Programmable GPIO
- Crystal-Free Operation Using Reference Over Reverse (RoR) Clocking
- Reference Clock Output for Image Sensor
- Built-in ADC and Temperature Sensor
- Compact 5mm x 5mm TQFN Side-Wettable Package with 0.5mm Pitch

## Simplified Applications Diagram



---

**TABLE OF CONTENTS**

---

General Description . . . . .	1
Applications . . . . .	1
Benefits and Features . . . . .	1
Simplified Applications Diagram . . . . .	2
Absolute Maximum Ratings . . . . .	10
Package Information . . . . .	10
32-Pin TQFN-SW (Side-Wettable) . . . . .	10
Electrical Characteristics . . . . .	11
Typical Operating Characteristics . . . . .	21
Pin Configuration . . . . .	22
Pin Descriptions . . . . .	22
Functional Diagrams . . . . .	26
Detailed Description . . . . .	27
Additional Documentation . . . . .	27
Recommended Operating Conditions, External Component Requirements and ESD Protection . . . . .	27
Functional Diagrams . . . . .	30
Introduction . . . . .	35
Product Overview . . . . .	35
Tunneling and Pixel Modes . . . . .	37
Video Pipeline . . . . .	38
Video Pipes in Pixel Mode . . . . .	38
RGB888 Video-Pattern Generator . . . . .	39
Tx Crossbar . . . . .	39
GMSL2 Packet Protocol . . . . .	39
Packet Protocol . . . . .	39
9b10b Link Encoding . . . . .	39
Tunneling Mode . . . . .	40
CRC Generation and Checking . . . . .	40
Control-Channel Retransmission on Error . . . . .	41
Automatic Repeat Request/Automatic Retransmission (ARQ) . . . . .	41
Forward-Error Correction . . . . .	41
Scheduler/Arbiter . . . . .	41
Bandwidth Sharing . . . . .	41
GMSL2 Physical Interface . . . . .	42
Link Speeds (Forward and Reverse) . . . . .	42
Cabling Options . . . . .	42
Selectable Line Rate . . . . .	42
Adaptive Equalization (AEQ) . . . . .	42
Echo Cancellation . . . . .	43

**TABLE OF CONTENTS (CONTINUED)**

Spread Spectrum .....	43
Line Fault.....	43
Coax Mode.....	43
STP Mode .....	44
Link Lock .....	45
GMSL2 Bandwidth Calculations.....	46
Control Channel and Side Channels .....	48
Main I <sup>2</sup> C/UART .....	48
I <sup>2</sup> C/UART CRC and Message Counter.....	48
I <sup>2</sup> C/UART CRC and Message Counter Options.....	49
Pass-Through I <sup>2</sup> C/UART.....	49
Serial Peripheral Interface (SPI).....	49
General-Purpose Input and Output.....	49
Delay in Non-Compensated Mode .....	50
Delay-Compensated Mode.....	50
Frame Sync .....	50
Functional Safety Features .....	50
Analog-to-Digital Converter .....	51
ADC Features .....	51
ADC Architecture.....	51
ADC Operation .....	52
ADC Control Interface .....	52
ADC Power-Up and Power-Down Sequences.....	53
ADC Overvoltage and Undervoltage Thresholds.....	53
ADC Round-Robin Sampling .....	53
ADC Application Diagrams for External Voltage Measurements .....	53
Measuring External Voltages Lower than V <sub>DDIO</sub> .....	53
Measuring Voltages Higher than V <sub>DDIO</sub> .....	54
Junction Temperature Monitoring and Overtemperature Alarm .....	54
Eye-Opening Monitor .....	55
Sleep Mode .....	55
Other Functions .....	55
Video PRBS .....	55
MIPI CSI-2 Input Interface .....	56
Lane Configurations and Data Rates .....	56
Clocking.....	56
CSI-2 Virtual Channel and Data Type Interleaving .....	56
Lane Deskew.....	56
Minimum Blanking.....	57

---

**TABLE OF CONTENTS (CONTINUED)**

---

MIPI End-to-End Packet Spacing .....	57
CFG Latch at Power-Up Pins .....	57
Multifunction Pin Assignments .....	59
Control-Channel Link and Power-Up .....	60
Device Reset .....	60
Clocking .....	60
GMSL Reference Clock .....	61
Reference Clock Input Jitter .....	61
Remote Clocking (RoR Channel) .....	61
Reference Over Reverse/RoR Channel .....	61
RoR Power-Up .....	61
Reference Clock/DPLL Generation .....	61
Spread-Spectrum Clocking .....	61
Error and Fault-Condition Monitoring .....	62
Power Supplies .....	62
Supply Sequencing .....	62
Overvoltage/Undervoltage .....	62
PCB Layout Guidelines for GMSL .....	63
Ground Plane .....	63
High-Speed GMSL Traces .....	63
Power-over-Coax Layout .....	63
Shielded-Twisted Pair Layout .....	64
Thermal Management .....	65
Unique Serial Number .....	65
Applications Information .....	66
Software Programming Model .....	66
Programming Notes .....	66
Control-Channel Programming .....	66
Conventional I <sup>2</sup> C/UART Control-Channel Programming .....	66
Host-to-Peripheral Main I <sup>2</sup> C and Pass-Through I <sup>2</sup> C Communication .....	66
I <sup>2</sup> C Write Packet Format .....	67
I <sup>2</sup> C Read Packet Format .....	67
Main I <sup>2</sup> C Host-to-GMSL2 Device Communication .....	67
Main UART .....	67
UART Base Mode .....	67
UART Bypass Mode .....	68
Switching Between UART Base and Bypass Modes .....	68
UART Frame Format .....	68
Synchronization Frame .....	69

---

**TABLE OF CONTENTS (CONTINUED)**

---

Acknowledge Frame . . . . .	69
Write Packet . . . . .	69
Read Packet . . . . .	70
I <sup>2</sup> C/UART Control-Channel Programming with Optional CRC and Message Counter . . . . .	70
I <sup>2</sup> C/UART CRC and Message Counter Options . . . . .	70
I <sup>2</sup> C Writes with CRC . . . . .	70
I <sup>2</sup> C Reads with CRC . . . . .	71
Message Counter Writes . . . . .	71
Message Counter Reads . . . . .	71
UART Writes with CRC . . . . .	71
UART Reads with CRC . . . . .	72
Typical Application Circuits . . . . .	74
Ordering Information . . . . .	75
Register Map . . . . .	76
MAX96717 . . . . .	76
Register Details . . . . .	94
Revision History . . . . .	246

---

**LIST OF FIGURES**

---

Figure 1. Serial Output Parameters . . . . .	30
Figure 2. GMSL2 Data Initialization Time . . . . .	30
Figure 3. I <sup>2</sup> C Timing Parameters . . . . .	31
Figure 4. GMSL2 Data Latency . . . . .	31
Figure 5. D-PHY LP Receiver Pulse . . . . .	31
Figure 6. D-PHY Data Clock Timing . . . . .	32
Figure 7. High-Speed Data Transmission in Bursts . . . . .	32
Figure 8. D-PHY High-Speed Skew Calibration . . . . .	33
Figure 9. Switching the Clock Lane Between Clock Transmission and Low-Power Mode . . . . .	33
Figure 10. Signaling and Contention Voltage Levels . . . . .	34
Figure 11. SPI Main-Mode Timing Parameters . . . . .	34
Figure 12. SPI Subordinate-Mode Timing Parameters . . . . .	35
Figure 13. Single Link . . . . .	36
Figure 14. Dual Link, Separate Data Type . . . . .	36
Figure 15. Dual Link, Data Aggregation . . . . .	37
Figure 16. Dual Link, Data Aggregation and Replication . . . . .	37
Figure 17. Pixel Mode . . . . .	38
Figure 18. Tunneling Mode . . . . .	38
Figure 19. Line Fault Detection Location Options . . . . .	43
Figure 20. Typical GMSL Link Application Circuit for Coax Cable . . . . .	44
Figure 21. Typical GMSL Link Application Circuit for Twisted Pair (Line Fault Pair #1: LMN0 and LMN1, Option #2) . .	45
Figure 22. GMSL2 Lock Time . . . . .	45
Figure 23. Video Frame Format for Bandwidth Calculation . . . . .	48
Figure 24. ADC Block Diagram . . . . .	52
Figure 25. Measuring External ADC Voltages < 3.6V . . . . .	54
Figure 26. Measuring External ADC Voltages Greater Than V <sub>DDIO</sub> . . . . .	54
Figure 27. MIPI CSI-2 Lane Mapping Options and Default Settings . . . . .	56
Figure 28. Video Timing . . . . .	57
Figure 29. End-to-End Packet Spacing . . . . .	57
Figure 30. Coax PoC Layout Example . . . . .	64
Figure 31. STP Layout Example . . . . .	65
Figure 32. I <sup>2</sup> C Write Packet Format . . . . .	67
Figure 33. I <sup>2</sup> C Read Packet Format . . . . .	67
Figure 34. UART Protocol for Base Mode . . . . .	68
Figure 35. UART Data Format for Base Mode . . . . .	69
Figure 36. UART Synchronization Frame . . . . .	69
Figure 37. UART Acknowledge Frame . . . . .	69
Figure 38. UART Write Packet Format . . . . .	70
Figure 39. UART Read Packet Format . . . . .	70

---

---

**LIST OF FIGURES (CONTINUED)**

---

Figure 40. I <sup>2</sup> C CRC Engine .....	70
Figure 41. I <sup>2</sup> C Multiple-Byte Write with CRC .....	71
Figure 42. I <sup>2</sup> C Single-Byte Write with CRC .....	71
Figure 43. I <sup>2</sup> C Multiple-Byte Read with CRC .....	71
Figure 44. I <sup>2</sup> C Single-Byte Read with CRC .....	71
Figure 45. UART CRC Engine .....	72
Figure 46. UART Multiple-Byte Write Transactions with CRC .....	72
Figure 47. UART Single-Byte Write Transaction with CRC .....	72
Figure 48. UART Multiple-Byte Read Transaction with CRC .....	72
Figure 49. UART Single-Byte Read Transaction with CRC .....	73
Figure 50. Typical Application Circuit .....	74

---

**LIST OF TABLES**

---

Table 1. Typical Maximum Cable Length vs. Attenuation . . . . .	1
Table 2. Recommended Operating Conditions . . . . .	27
Table 3. External Component Requirements . . . . .	27
Table 4. ESD Protection . . . . .	28
Table 5. Coax Mode Line Fault Configuration Options . . . . .	44
Table 6. STP Mode Line Fault Configuration Options for LMN0/LMN1 . . . . .	44
Table 7. Forward- and Reverse-Link Bandwidth Utilization . . . . .	47
Table 8. SPI Latching Edge and Speed . . . . .	49
Table 9. Typical GPIO Delays for Forward-Link and Reverse-Link Transmission . . . . .	50
Table 10. ADC Voltage Divider Settings . . . . .	53
Table 11. CFG0 Input Map . . . . .	58
Table 12. CFG1 Input Map . . . . .	58
Table 13. MFP Pin Function Map . . . . .	59
Table 14. Control- and Side-Channel Typical Rise and Fall Times. . . . .	60

## Absolute Maximum Ratings

(All voltages with respect to ground) .....	-0.3V to +3.9V	D_P/N, CKP/N .....	-0.3V to +1.35V
$V_{DDIO}$ .....	-0.3V to +3.9V	$X_{RES}, X_2$ .....	-0.3V to ( $V_{DD18} + 0.3V$ )
$V_{DD18}$ .....	-0.3V to +2.0V	All Other Pins ( <a href="#">Note B</a> ) .....	-0.3V to ( $V_{DDIO} + 0.3V$ )
$V_{DD}$ .....	-0.3V to +2.0V	Continuous Power Dissipation ( $T_A = +70^\circ C$ ) TQFN & TQFN-SW, Multilayer Board, Derate 34.5mW/ $^\circ C$ above +70 $^\circ C$ .....	1896mW
$CAP_{VDD}$ .....	-0.3V to +1.2V	Storage Temperature Range .....	-40 $^\circ C$ to +150 $^\circ C$
$SIO_{(Active\ State)}$ ( <a href="#">Note A</a> ) .....	-0.3V to ( $CAP_{VDD} + 0.3V$ )	Soldering Temperature (Reflow) .....	+260 $^\circ C$
$SIO_{(Inactive\ State)}$ ( <a href="#">Note A</a> ) .....	-0.3V to ( $CAP_{VDD} + 0.3V$ )		

**Note A:** Active state means the device is powered-up and not in Sleep or Power-down modes. Inactive means the device is not powered-up or powered-up in Sleep or Power-down mode.

**Note B:**  $V_{DDIO}$  = specified maximum voltage or 3.9V, whichever is lower.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Package Information

### 32-Pin TQFN-SW (Side-Wettable)

Package Code	T3255Y (Click on Land Pattern Number link below for additional details.)
Outline Number	<a href="#">21-100156</a>
Land Pattern Number	<a href="#">90-100067</a>
<b>Thermal Resistance, Four-Layer Board</b>	
Junction to Ambient ( $\theta_{JA}$ )	29 $^\circ C/W$
Junction to Case ( $\theta_{JC}$ )	1.7 $^\circ C/W$

For the latest package outline information and land patterns (footprints), go to [www.analog.com/packages](http://www.analog.com/packages). Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board in still air. For detailed information on package thermal considerations, refer to [www.analog.com/thermal-tutorial](http://www.analog.com/thermal-tutorial).

## Electrical Characteristics

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^\circ C$  to  $+105^\circ C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^\circ C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>DC ELECTRICAL CHARACTERISTICS / GMSL2 FORWARD-CHANNEL SERIAL INPUTS/OUTPUTS (SIOP, SION)—SEE <a href="#">Figure 1</a></b>						
Output-Voltage Swing (Single-Ended)	$V_O$	$R_L = 100\Omega \pm 1\%$ , $(V_{OH} - V_{OL})$ for both outputs	300	400	500	mV
Output-Voltage Swing (Differential)	$V_{ODT}$	$R_L = 100\Omega \pm 1\%$ , peak-to-peak differential voltage	600	800	1000	mV
Change in $V_{OD}$ Between Complementary Output States	$\Delta V_{OD}$	$R_L = 100\Omega \pm 1\%$ , $ V_{OD(H)} - V_{OD(L)} $		25		mV
Differential Output Offset Voltage	$V_{OS}$	$R_L = 100\Omega \pm 1\%$ , offset voltage in each output state	0.4	0.5	0.6	V
Change in $V_{OS}$ Between Complementary Output States	$\Delta V_{OS}$	$R_L = 100\Omega \pm 1\%$ , $ V_{OS(H)} - V_{OS(L)} $		25		mV
Termination Resistance (Internal)	$R_O$	Any pin to $V_{DD18}$	45	50	55	$\Omega$
<b>DC ELECTRICAL CHARACTERISTICS / D-PHY HS RECEIVER</b>						
Common-Mode Voltage HS Receive Mode	$V_{CMRX(DC)}$		70		330	mV
Differential Input High Threshold	$V_{IDTH}$			40		mV
Differential Input Low Threshold	$V_{IDTL}$		-40			mV
Single-Ended Input High Voltage	$V_{IHHS}$			460		mV
Single-Ended Input Low Voltage	$V_{ILHS}$		-40			mV
Single-Ended Threshold for HS Termination Enable	$V_{TERM-EN}$			450		mV
Differential Input Impedance	$Z_{ID}$		80	100	125	$\Omega$
<b>DC ELECTRICAL CHARACTERISTICS / D-PHY LP RECEIVER—SEE <a href="#">Figure 5</a></b>						
Logic 1 Input Voltage	$V_{IH}$		740			mV
Logic 0 Input Voltage, Not in ULP State	$V_{IL}$			550		mV
Input Hysteresis	$V_{HYST}$		25			mV
Pin Leakage Current	$I_{LEAK}$	-0.05V to 1.35V	-10	10		$\mu A$
<b>DC ELECTRICAL CHARACTERISTICS / I/O PINS (GPIO)</b>						
High-Level Input Voltage	$V_{IH}$		$0.7 \times V_{DDIO}$			V
Low-Level Input Voltage	$V_{IL}$			$0.3 \times V_{DDIO}$		V

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
High-Level Output Voltage	$V_{OH}$	$I_{OH} = -4mA$	$V_{DDIO} - 0.4$			V
Low-Level Output Voltage	$V_{OL}$	$I_{OL} = 4mA$		0.4		V
Input Current	$I_{IN}$	$V_{IN} = 0$ to $V_{DDIO}$ . All pullup/pulldown devices disabled.		1		$\mu A$
Input Capacitance	$C_{IN}$	Each pin		5		pF
Input Pullup/Pulldown Resistance	$R_{IN}$	40k $\Omega$ enabled		40		k $\Omega$
		1M $\Omega$ enabled		1		M $\Omega$
<b>DC ELECTRICAL CHARACTERISTICS / OPEN-DRAIN PINS</b>						
High-Level Input Voltage	$V_{IH}$		$0.7 \times V_{DDIO}$			V
Low-Level Input Voltage	$V_{IL}$			$0.3 \times V_{DDIO}$		V
Low-Level Open-Drain Output Voltage	$V_{OL}$	$I_{OL} = 4mA$		0.4		V
Input Current	$I_{IN}$	$V_{IN} = 0$ to $V_{DDIO}$ . All pullup/pulldown devices disabled.		1		$\mu A$
Input Capacitance	$C_{IN}$	Each pin		3		pF
Internal Pullup Resistor	$R_{PU}$	40k $\Omega$ enabled		40		k $\Omega$
		1M $\Omega$ enabled		1		M $\Omega$
<b>DC ELECTRICAL CHARACTERISTICS / PWDNB INPUT</b>						
High-Level Input Voltage	$V_{IH}$		$0.7 \times V_{DDIO}$			V
Low-Level Input Voltage	$V_{IL}$			$0.3 \times V_{DDIO}$		V
Input Current	$I_{IN}$	$V_{IN} = 0$ to $V_{DDIO}$		6		$\mu A$
Input Capacitance	$C_{IN}$			3		pF
Internal Pulldown Resistor	$R_{PD}$			1		M $\Omega$
<b>DC ELECTRICAL CHARACTERISTICS / PUSH-PULL OUTPUTS (GPIO)</b>						
High-Level Output Voltage	$V_{OH}$	$I_{OH} = -4mA$	$V_{DDIO} - 0.4$			V
Low-Level Output Voltage	$V_{OL}$	$I_{OL} = 4mA$		0.4		V
<b>DC ELECTRICAL CHARACTERISTICS / LINE FAULT DETECTION INPUT (LMN0, LMN1)</b>						
Open-Pin Voltage	$V_{O0}$	LMN0		1.25		V
	$V_{O1}$	LMN1		0.75		
<b>DC ELECTRICAL CHARACTERISTICS / REFERENCE CLOCK INPUT (CRYSTAL) (X1/OSC, X2)</b>						
X1/OSC Input Capacitance	$C_{IN\_X1}$			3		pF
X2 Input Capacitance	$C_{IN\_X2}$			1		pF

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^\circ C$  to  $+105^\circ C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^\circ C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Internal X2 Limit Resistor	$R_{LIM}$			1.2		$k\Omega$
Internal Feedback Resistor	$R_{FB}$			10		$k\Omega$
Transconductance	$g_m$			28		$mA/V$
<b>DC ELECTRICAL CHARACTERISTICS / REFERENCE CLOCK REQUIREMENTS (EXTERNAL INPUT ON X1/OSC, X2 FLOATING)</b>						
High-Level Input Voltage	$V_{IH}$		0.9			V
Low-Level Input Voltage	$V_{IL}$			0.4		V
Input Resistance	$R_{IN}$		10			$k\Omega$
X1 Input Capacitance	$C_{IN\_X1}$		3			$pF$
<b>DC ELECTRICAL CHARACTERISTICS / MONITOR ADC</b>						
Resolution			10			bits
Input-Voltage Range	$V_{IN}$	Using internal reference, no divider, no internal buffer.	0.0 to $V_{REF}$			V
		Using internal reference, 2:1 divider, no internal buffer, ADC0, ADC1, ADC2 pins.	0.0 to lower of $(2 \times V_{REF})$ or $V_{DDIO}$			
		Using internal reference, 3:1 divider, no internal buffer, ADC0, ADC1, ADC2 pins.	0.0 to lower of $(3 \times V_{REF})$ or $V_{DDIO}$			
		Using internal reference, 4:1 divider, no internal buffer, ADC0, ADC1, ADC2 pins.	0.0 to $V_{DDIO}$			
		No divider, buffered input	0.1 to $V_{REF}$			
Input Resistance	$R_{IN}$	Input configured for 2:1 voltage division	60			$k\Omega$
		Input configured for 3:1 voltage division	45			
		Input configured for 4:1 voltage division	40			
		No divider, buffered input	> 5			$M\Omega$
Divider Ratio	DR	Divider Ratio	GPIO /1 (no divider)	1		V/V
			GPIO /2	2		
			GPIO /3	3		
			GPIO /4	4		
			$V_{DDIO}/4$ Input	4.017		
			$V_{DD18}/2$ Input	2.009		
			CAP_VDD/2 Input	2.009		

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Total Unadjusted Error	TUE	External $V_{REF} = 1.25V$ , no divider, buffered input	External ADC (0-2) voltage = $0.1V$	-10	+10	mV
		External ADC (0-2) voltage = $0.625V$	-12	+12		
		External ADC (0-2) voltage = $1.2V$	-15	+15		
		Internal Reference ( $V_{REF} = 1.25V$ ), no divider, buffered input	External ADC (0-2) voltage = $0.1V$	-10	+10	
		External ADC (0-2) voltage = $0.625V$	-12	+12		
		External ADC (0-2) voltage = $1.2V$	-15	+15		
		Internal Reference ( $V_{REF} = 1.25V$ ), GPIO/2, buffered input	External ADC (0-2)	-15	+15	
		Internal Reference ( $V_{REF} = 1.25V$ ), GPIO/3, buffered input	External ADC (0-2)	-30	+30	
		Internal Reference ( $V_{REF} = 1.25V$ ), GPIO/4, buffered input	External ADC (0-2)	-42	+42	
		Internal Reference ( $V_{REF} = 1.25V$ ), buffered input	$V_{DDIO}/4$ Input	-40	+40	
		$V_{DD18}/2$ Input	-15	+15		
		CAP_VDD/2 Input	-15	+15		
Reference-Voltage Tolerance for 1% ADC Accuracy	$V_{REF\_TOL}$	External $V_{REF} = 1.25V$ , no divider, buffered input		$\pm 1$		mV
<b>DC ELECTRICAL CHARACTERISTICS / TEMPERATURE MONITOR (<a href="#">Note 2</a>)</b>						
Measurable Range	T1		-40	+105		°C
Accuracy	ERROR <sub>T1</sub>	-40°C to +105°C		$\pm 3$		°C
Measurable Range	T2		+105	125		°C
Accuracy	ERROR <sub>T2</sub>	+105°C to +125°C	-2.5	+2.5		°C
<b>DC ELECTRICAL CHARACTERISTICS / POWER SUPPLY CURRENTS (GMSL2 MODE)</b>						
Supply Current ( <a href="#">Note 3</a> )	$I_{DD}$	RGB888, 6Gbps, 4-Lane CSI-2 input, 1.3Gbps per lane	$V_{DD18} = 1.9V$	46	53	mA
			$V_{DD} = 1.05V$	98	223	
			$V_{DD} = 1.26V$	98	232	
Maximum $V_{DDIO}$ Supply Current ( <a href="#">Note 4</a> )	$I_{DDIO}$	Per toggling GPIO, $C_L = 20pF$	$V_{DDIO} = 1.9V$	44		$\mu A/MHz$
			$V_{DDIO} = 3.6V$	81		

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
<b>DC ELECTRICAL CHARACTERISTICS / POWER-DOWN CURRENT</b>							
Maximum Power-Down Current	$I_{DD}$	$V_{DDIO}$ at $3.6V$	$T_A = +25^{\circ}C$	6.4			$\mu A$
			$T_A = +105^{\circ}C$	6.1			
		$V_{DD18}$ at $1.9V$	$T_A = +25^{\circ}C$	0.5			
			$T_A = +105^{\circ}C$	1.9			
		$V_{DD}$ at $1.26V$	$T_A = +25^{\circ}C$	3			
			$T_A = +105^{\circ}C$	48			
<b>DC ELECTRICAL CHARACTERISTICS / SLEEP CURRENT</b>							
Maximum Sleep Current	$I_{DD}$	$V_{DDIO}$ at $3.6V$	$T_A = +25^{\circ}C$	7.7			$\mu A$
			$T_A = +105^{\circ}C$	7.4			
		$V_{DD18}$ at $1.9V$	$T_A = +25^{\circ}C$	0.5			
			$T_A = +105^{\circ}C$	5			
		$V_{DD}$ at $1.26V$	$T_A = +25^{\circ}C$	6.2			
			$T_A = +105^{\circ}C$	49			
<b>AC ELECTRICAL CHARACTERISTICS / FORWARD CHANNEL SWITCHING CHARACTERISTICS</b>							
Serial-Output Rise Time	$t_R$	20% to 80%, $V_{OD} = 800mV$ differential, $R_L = 100\Omega$ , 500mV single-ended $R_L = 50\Omega$		50			ps
Serial-Output Fall Time	$t_F$	80% to 20%, $V_{OD} = 800mV$ differential, $R_L = 100\Omega$ , 400mV single-ended $R_L = 50\Omega$		50			ps
Total Serial-Output Jitter	$t_{TSOJ}$	PRBS7, single-ended or differential		0.15			UI (p-p)
Deterministic Serial-Output Jitter	$t_{DSOJ}$	PRBS7, single-ended or differential		0.10			UI (p-p)
Lock Time	$t_{LOCK}$	Time from deassertion of PWDNB until $LOCK = '1'$ . See <a href="#">Figure 22</a>		45			ms
Maximum Video Latency	$t_{VL}$	Time from CSI-2 input to SIO $\pm$ output in GMSL2 packet. See <a href="#">Figure 4</a> .	Pixel mode	$120 \times t_{PCLK}$			s
			Tunneling mode 6G	$30 \times t_{PCLK} + 1\mu s$			
			Tunneling mode 3G	$30 \times t_{PCLK} + 2\mu s$			
PWDNB Hold Time	$t_{HOLD\_PWDNB}$	The minimum duration PWDNB must be held LOW to reset the device.		1			ms
Data Initialization Time	$t_{PU}$	Time from initial data being applied to CSI-2 until it appears at SIOP, SION. Assumes link is already established. See <a href="#">Figure 2</a>		2			ms
GPI-to-GPO Delay	$t_{GPD1}$	Delay compensated mode.		10			$\mu s$
	$t_{GPD2}$	Non-delay compensated mode.		3			

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
GPI-GPO Skew Reverse Path	$t_{SKEW}$	Delay compensated mode.		7		ns
<b>AC ELECTRICAL CHARACTERISTICS / D-PHY HS RECEIVER</b>						
Common-Mode Interference Beyond 450MHz	$\Delta V_{CMRX(HF)}$	<a href="#">Note 2</a> , <a href="#">Note 5</a> , <a href="#">Note 8</a>	Data rate > 1.5Gbps		50	mV
Common-Mode Interference 50MHz–450MHz	$\Delta V_{CMRX(LF)}$	<a href="#">Note 2</a> , <a href="#">Note 5</a> , <a href="#">Note 6</a>	Data rate > 1.5Gbps	-25	25	mV
Common-Mode Reflection Coefficient	$S_{ccRX}$	450MHz < $f$ < 1.875GHz		-5		dB
Differential-Mode Reflection Coefficient	$S_{ddRX}$	$f < 20MHz$		-22.5		dB
		$f = 1.25GHz$		-12		
		$f = 1.875GHz$		-9.7		
<b>AC ELECTRICAL CHARACTERISTICS / D-PHY LP RECEIVER—SEE <a href="#">Figure 5 (Note 2)</a></b>						
Input Pulse Rejection	$e_{SPIKE}$	See <a href="#">Figure 5</a>		300		V*ps
Minimum Pulse Width Response	$t_{MIN-RX}$	See <a href="#">Figure 5</a>	20			ns
Peak Interference Amplitude	$V_{INT}$			200		mV
Interference Frequency	$f_{INT}$		450			MHz
<b>AC ELECTRICAL CHARACTERISTICS / D-PHY DATA CLOCK TIMING—SEE <a href="#">Figure 6 (Note 2)</a></b>						
UI Instantaneous	$UI_{INST}$		0.4	12.5		ns
UI Variation	$\Delta UI$	$UI \geq 1ns$ , within a single burst	-10%	+10%		UI
		$0.667ns < UI < 1ns$ , within a single burst	-5%	5%		
Data to Clock Setup Time	$t_{SETUP[RX]}$	< 1.0Gbps	0.15			$UI_{INST}$
		> 1.0Gbps	0.2			
Data to Clock Hold Time	$t_{HOLD[RX]}$	< 1.0Gbps	0.15			$UI_{INST}$
		> 1.0Gbps	0.2			
Static Data to Clock Skew	$T_{SKEW[RX]}_{static}$	> 1.5Gbps	-0.2	0.2		$UI_{INST}$
Dynamic Data to Clock Skew Window Rx Tolerance	$T_{SETUP[RX]} + T_{HOLD[RX]}_{Dynamic}$	> 1.5Gbps	0.50			$UI_{INST}$
<b>AC ELECTRICAL CHARACTERISTICS / D-PHY GLOBAL OPERATION TIMING (<a href="#">Note 2</a>)</b>						
Time Interval When the HS Receiver Ignores Any Clock Lane HS Transitions, Starting from the Beginning of $T_{CLK-PREPARE}$ .	$t_{CLK-SETTLE}$	See <a href="#">Figure 9</a>	95	300		ns

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Time for the Clock Lane Receiver to Enable the HS Line Termination, Starting from the Time Point When Dn Crosses $V_{IL,MAX}$ .	tCLK-TERM-EN	See <a href="#">Figure 9</a>	Time for Dn to reach $V_{TERM-EN}$	38		ns
Time for the Data Lane Receiver to Enable the HS Line Termination, Starting from the Time Point When Dn Crosses $V_{IL,MAX}$ .	tD-TERM-EN	See <a href="#">Figure 7</a> and <a href="#">Figure 8</a>	Time for Dn to reach $V_{TERM-EN}$	35ns + 4*UI		ns
Time Interval When the HS Receiver Ignores Any Data Lane HS Transitions, Starting from the Beginning of T <sub>HS-PREPARE</sub> .	tHS-SETTLE	See <a href="#">Figure 7</a> and <a href="#">Figure 8</a>	85 + 6*UI	145 + 10*UI		ns

### AC ELECTRICAL CHARACTERISTICS / I<sup>2</sup>C/UART PORT TIMING—SEE [Figure 3](#)

Output Fall Time	t <sub>F</sub>	70% to 30%, $C_L = 20pF$ to $100pF$ , $1k\Omega$ pullup to $V_{DDIO}$ . ( <a href="#">Note 2</a> )	20 x $V_{DDIO}/5.5V$	150		ns
I <sup>2</sup> C/UART Wake Time		From power-up or rising edge of PWDNB to local register access. For remote register access, I <sup>2</sup> C/UART wake time is the same as lock time (t <sub>LOCK</sub> ).		1.1		ms

### AC ELECTRICAL CHARACTERISTICS / I<sup>2</sup>C TIMING—SEE [Figure 3](#)

SCL Clock Frequency	f <sub>SCL</sub>	Low f <sub>SCL</sub> range : (I2CMSTBT = 010, I2CSLVSH = 10)	9.6	100	kHz
		Mid f <sub>SCL</sub> range : (I2CMSTBT = 101, I2CSLVSH = 01)	100	400	
		High f <sub>SCL</sub> range : (I2CMSTBT = 111, I2CSLVSH = 00)	400	1000	
Start Condition Hold Time	t <sub>HD:STA</sub>	f <sub>SCL</sub> range, Low	4		μs
		f <sub>SCL</sub> range, Mid	0.6		
		f <sub>SCL</sub> range, High	0.26		
Low Period of SCL Clock	t <sub>LOW</sub>	f <sub>SCL</sub> range, Low	4.7		μs
		f <sub>SCL</sub> range, Mid	1.3		
		f <sub>SCL</sub> range, High	0.5		
High Period of SCL Clock	t <sub>HIGH</sub>	f <sub>SCL</sub> range, Low	4		μs
		f <sub>SCL</sub> range, Mid	0.6		
		f <sub>SCL</sub> range, High	0.26		
Repeated Start Condition Setup Time	t <sub>SU:STA</sub>	f <sub>SCL</sub> range, Low	4.7		μs
		f <sub>SCL</sub> range, Mid	0.6		
		f <sub>SCL</sub> range, High	0.26		

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Data Hold Time	$t_{HD:DAT}$	$f_{SCL}$ range, Low	0			ns
		$f_{SCL}$ range, Mid	0			
		$f_{SCL}$ range, High	0			
Data Setup Time	$t_{SU:DAT}$	$f_{SCL}$ range, Low	250			ns
		$f_{SCL}$ range, Mid	100			
		$f_{SCL}$ range, High	50			
Setup Time for Stop Condition	$t_{SU:STO}$	$f_{SCL}$ range, Low	4			$\mu s$
		$f_{SCL}$ range, Mid	0.6			
		$f_{SCL}$ range, High	0.26			
Bus Free Time	$t_{BUF}$	$f_{SCL}$ range, Low	4.7			$\mu s$
		$f_{SCL}$ range, Mid	1.3			
		$f_{SCL}$ range, High	0.5			
Data Valid Time	$t_{VD:DAT}$	$f_{SCL}$ range, Low		3.45		$\mu s$
		$f_{SCL}$ range, Mid		0.9		
		$f_{SCL}$ range, High		0.45		
Data Valid Acknowledge Time	$t_{VD:ACK}$	$f_{SCL}$ range, Low		3.45		$\mu s$
		$f_{SCL}$ range, Mid		0.9		
		$f_{SCL}$ range, High		0.45		
Pulse Width of Spikes Suppressed	$t_{SP}$	$f_{SCL}$ range, Low		50		ns
		$f_{SCL}$ range, Mid		50		
		$f_{SCL}$ range, High		50		
Capacitive Load on Each Bus Line	$C_B$			100		pF

### AC ELECTRICAL CHARACTERISTICS / SPI MAIN—SEE [Figure 11 \(Note 7\)](#)

Programmable Operating Frequency Range	$f_{MCK}$	( <a href="#">Note 2</a> )	0.588	25	MHz
SCLK Period	$t_{MCK}$		$1/f_{MCK}$		ns
SCLK Output Pulse-Width High/Low	$t_{MCH}, t_{MCL}$	$C_L = 5pF$ ( <a href="#">Note 2</a> )	$t_{MCK}/2 - 3$	$t_{MCK}/2$	ns
MOSI Data Output Delay	$t_{MOD}$	After SCLK falling edge ( <a href="#">Note 2</a> )	-6	3	ns
MISO Input Setup Time	$t_{MIS}$	Before programmed sampling edge ( <a href="#">Note 2</a> )	13.5		ns
MISO Input Hold Time	$t_{MIH}$	After programmed sampling edge ( <a href="#">Note 2</a> )	-2		ns

### AC ELECTRICAL CHARACTERISTICS / SPI SUBORDINATE—SEE [Figure 12 \(Note 7\)](#)

Operating Frequency	$f_{SCK}$	( <a href="#">Note 2</a> )	50	MHz	
SCLK Period	$t_{SCK}$		$1/f_{SCK}$	ns	
MISO Data Output Delay	$t_{SOD}$	After SCLK falling edge ( <a href="#">Note 2</a> )	2	11.3	ns

## Electrical Characteristics (continued)

( $V_{DD18} = 1.7V$  to  $1.9V$ ,  $V_{DD} = 0.95V$  to  $1.26V$ ,  $V_{DDIO} = 1.7V$  to  $3.6V$ ,  $T_A = -40^{\circ}C$  to  $+105^{\circ}C$ , EP connected to PCB ground, typical values are at  $V_{DD18} = V_{DDIO} = 1.8V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^{\circ}C$ , unless otherwise noted. ([Note 1](#))

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
MOSI Input Setup Time	$t_{SIS}$	Before SCLK rising edge ( <a href="#">Note 2</a> )	5			ns
MOSI Input Hold Time	$t_{SIH}$	After SCLK rising edge ( <a href="#">Note 2</a> )	3			ns
<b>AC ELECTRICAL CHARACTERISTICS / MONITOR ADC</b>						
Conversion Time			430			μs
ADC Setup Time	$T_{ADC\_SETUP}$	ADC ready after power-up enable. Excluding external $V_{REF}$ power-up time.	30			μs
ADC Clock	$f_{ADCCLK}$		2.5			MHz
<b>AC ELECTRICAL CHARACTERISTICS / REFERENCE CLOCK INPUT REQUIREMENTS (CRYSTAL) (X1,X2) (<a href="#">Note 2</a>)</b>						
Frequency		Fundamental mode only	25			MHz
Frequency Stability + Frequency Tolerance	$f_{TN}$			±200		ppm
<b>AC ELECTRICAL CHARACTERISTICS / REFERENCE CLOCK REQUIREMENTS (EXTERNAL CLOCK INPUT ON X1, X2 FLOATING) (<a href="#">Note 2</a>)</b>						
Frequency	$f_{REF}$		25			MHz
Frequency Stability + Frequency Tolerance	$f_{TN}$			±200		ppm
Input Jitter	$t_{JIN}$	6Gbps/187.5Mbps, sinusoidal jitter < 1MHz (rising edge), downstream deserializer using crystal reference	600			ps (p-p)
Duty Cycle	$T_{DUTY}$		40	60		%
Maximum Rise Time	$t_R$	10% to 90%	5			ns
Maximum Fall Time	$t_F$	80% to 20%	4			ns
<b>AC ELECTRICAL CHARACTERISTICS / REFERENCE CLOCK OUTPUT (RCLKOUT)</b>						
Frequency	$f_{REF}$	Crystal or reference clock input.	25			MHz
Rise Time	$t_R$	20% to 80%, $C_L = 10pF$ , 12.5MHz (25MHz divided by 2), ( <a href="#">Note 4</a> )	2			ns
Fall Time	$t_F$	80% to 20%, $C_L = 10pF$ , 12.5MHz (25MHz divided by 2), ( <a href="#">Note 4</a> )	2			ns
Jitter	$t_J$	$C_L = 10pF$ , Rising or falling edge, 12.5MHz (25MHz divided by 2)	100			ps (p-p)
<b>AC ELECTRICAL CHARACTERISTICS / DIGITAL PLL OUTPUT (DPLL_OUT)</b>						
Frequency	$f_{DPLL}$	Maximum		75		MHz
		Minimum		1		
Rise Time	$t_R$	20% to 80%, $C_L = 10pF$ ( <a href="#">Note 4</a> )	2			ns
Fall Time	$t_F$	80% to 20%, $C_L = 10pF$ ( <a href="#">Note 4</a> )	2			ns
Jitter	$t_J$	Rising or falling edge, deterministic jitter + 14 x random jitter, 27MHz. ( <a href="#">Note 4</a> )	360			ps (p-p)

**Note 1:** Limits are 100% tested at  $T_A = +105^{\circ}C$ . Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization.

**Note 2:** Not production tested. Guaranteed by design and characterization.

**Note 3:** Color bar pattern. Maximum supply currents are measured at indicated supply voltages. Typical supply currents are measured at the typical supply voltages.

**Note 4:** MFP pin speed programmed to fastest setting. See the Multifunction Pin Assignments section.

**Note 5:** Excluding static ground shift of 50mV.

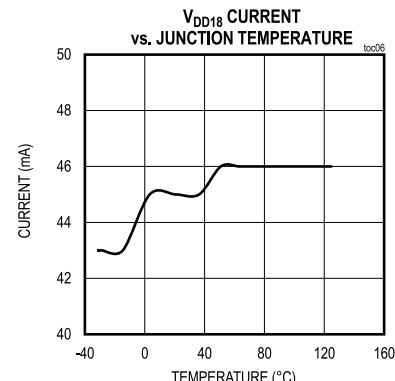
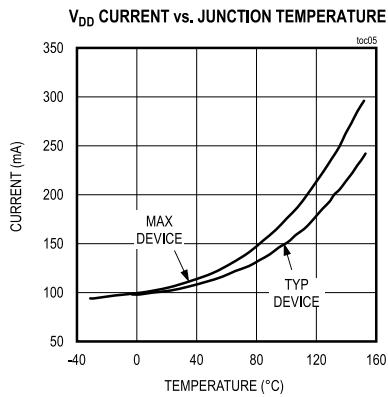
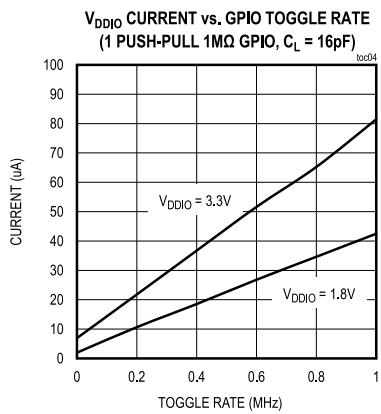
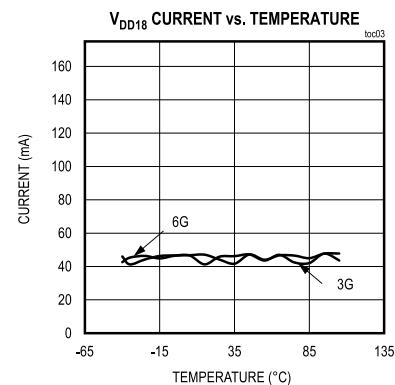
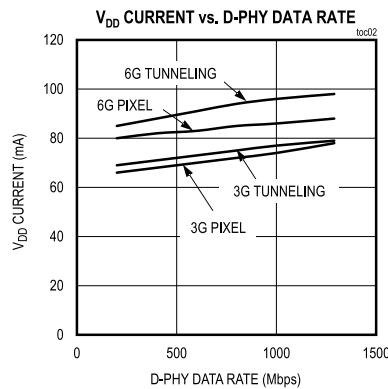
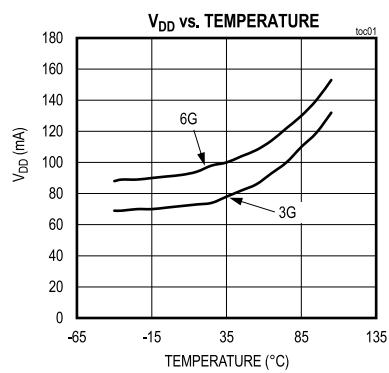
**Note 6:** Voltage difference compared to the DC average common-point potential.

**Note 7:** Measured at 50MHz, for  $V_{DDIO} = 1.7V$  to  $2.24V$ , Slew speed (PIOxxx\_SLEW[1:0] = 00) for MOSI, MISO, SCLK, RO, and BNE pins, and set to (PIOxxx\_SLEW[1:0] = 01) for  $V_{DDIO} = 2.2$  to  $3.6V$ .

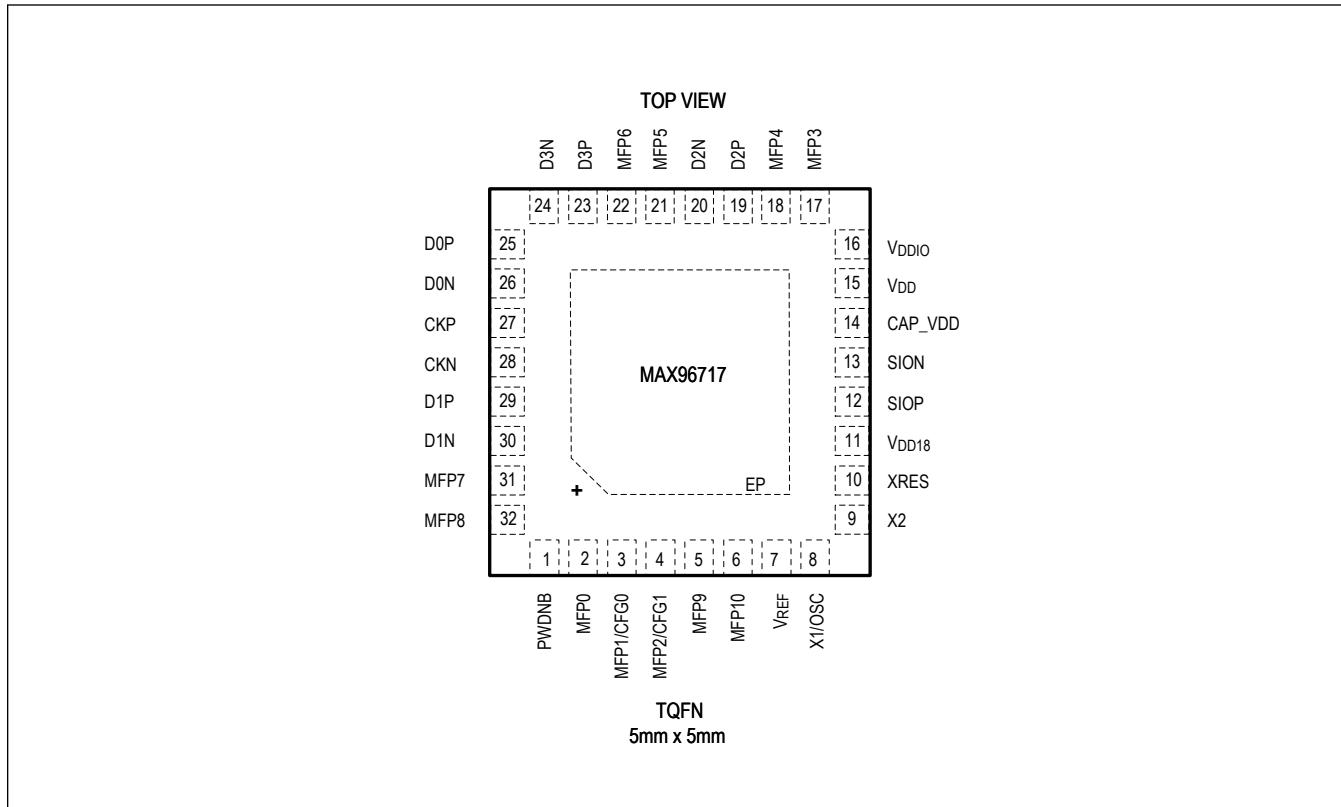
**Note 8:**  $\Delta V_{CMRX(HF)}$  is the peak amplitude of a sine-wave superimposed on the receiver inputs.

## Typical Operating Characteristics

( $V_{DD18} = 1.8V$ ,  $V_{DDIO} = 3.3V$ ,  $V_{DD} = 1.0V$ ,  $T_A = +25^\circ C$  unless otherwise noted. PRBS24 data, 6Gbps forward rate: 187.5Mbps reverse rate, 1.3Gbps per lane on 4 lanes DPHY. 3Gbps forward rate: 1.3Gbps per lane on 2 lanes DPHY.)



## Pin Configuration



## Pin Descriptions

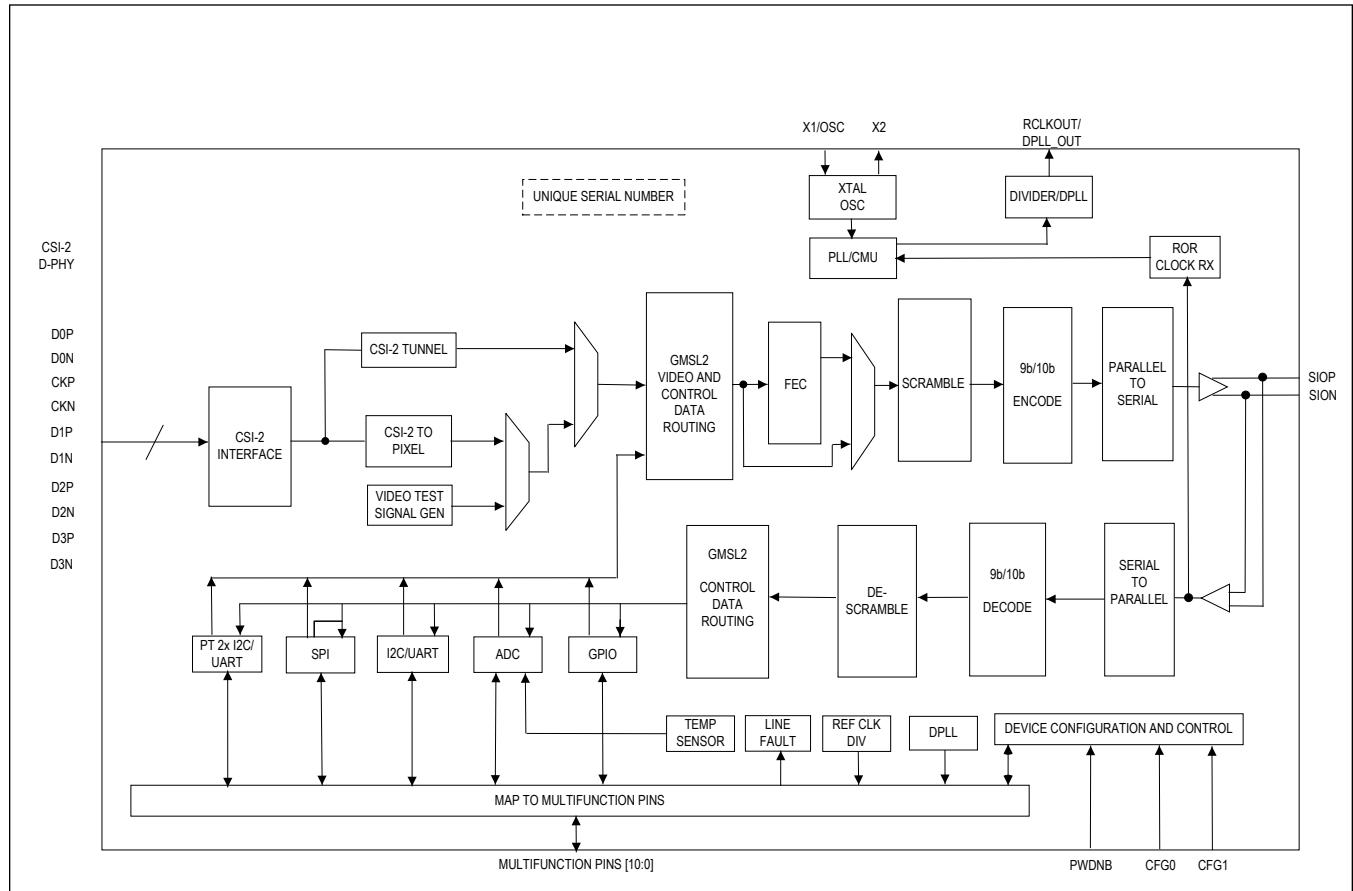
PIN	NAME	FUNCTION MODE		FUNCTION
		GMSL2		
<b>GMSL2 SERIAL LINK</b>				
12	SIOP	SIOP		Noninverted Coax/Twisted-Pair Serial-Data Input/Output.
13	SION	SION		Inverted Twisted-Pair Serial-Data Input/Output. Terminate with 100nF in series with 50Ω to ground in Coax mode.
<b>CSI-2 INTERFACE</b>				
19	D2P	D2P		CSI-2 Data Lane 2 Noninverted Input. Leave open if unused.
20	D2N	D2N		CSI-2 Data Lane 2 Inverted Input. Leave open if unused.
23	D3P	D3P		CSI-2 Data Lane 3 Noninverted Input. Leave open if unused.
24	D3N	D3N		CSI-2 Data Lane 3 Inverted Input. Leave open if unused.
25	D0P	D0P		CSI-2 Data Lane 0 Noninverted Input. Leave open if unused.
26	D0N	D0N		CSI-2 Data Lane 0 Inverted Input. Leave open if unused.
27	CKP	CKP		CSI-2 Clock Lane Noninverted Input.
28	CKN	CKN		CSI-2 Clock Lane Inverted Input.
29	D1P	D1P		CSI-2 Data Lane 1 Noninverted Input. Leave open if unused.
30	D1N	D1N		CSI-2 Data Lane 1 Inverted Input. Leave open if unused.

PIN	NAME	FUNCTION MODE	FUNCTION
		GMSL2	
<b>MULTIFUNCTION PINS (*) INDICATES DEFAULT STATE AFTER POWER-UP (LEAVE OPEN IF NOT USED)</b>			
2	MFP0	GPIO0* SCLK	GPIO0: Configurable General-Purpose Input or Output. Power-up default is high impedance with a 1MΩ pulldown resistor. SCLK: SPI Clock. When configured as SPI main, push-pull clock output. When configured as subordinate, clock input with 1MΩ pulldown to ground.
3	MFP1/CFG0	CFG0 GPO1* SS1 BNE	CFG0: Configuration Pin. Voltage at pin sets device modes, which are latched at power-up. Connect to a resistor divider between V <sub>DDIO</sub> and ground. See <a href="#">Table 11</a> . GPO1: General-Purpose Output. Power-up default is high impedance. SS1: When Configured as SPI Main, SS1 is a Subordinate Select Output. BNE: When Configured as SPI Subordinate, BNE Output Indicates SPI Data is Available.
4	MFP2/CFG1	CFG1 GPO2* RCLKOUT(Alt) DPLL_OUT(Alt)	CFG1: Configuration Pin. Voltage at pin sets device modes, which are latched at power-up. Connect to a resistor divider between V <sub>DDIO</sub> and ground. See <a href="#">Table 12</a> . GPO2: General-Purpose Output. Power-up default is high impedance. RCLKOUT (Alternate): 25MHz Frequency Reference Output. Divide by 2 or 4 available. DPLL_OUT (Alternate): Digital PLL Output. Programmable from 1MHz to 75MHz.
17	MFP3	GPIO3* ADC0 LOCK ERRB	GPIO3: Configurable General-Purpose Input or Output. Power-up default is high impedance with a 1MΩ pulldown resistor and GPIO disabled. ADC0: Voltage Monitor Analog Input. LOCK: High Indicates GMSL Link is Locked with Correct Serial Word Boundary Alignment. ERRB: Line Fault or Error Output. Goes low when a line fault on LMN0 or LMN1, a CSI-2 ECC/CRC or other error (as configured) is detected.
18	MFP4	GPIO4* RCLKOUT DPLL_OUT SS2 RO	GPIO4: Configurable General-Purpose Input or Output. Power-up default is high impedance with a 1MΩ pulldown resistor and GPIO disabled. RCLKOUT: 25MHz Frequency Reference Output. Divide by 2 or 4 available. DPLL_OUT: Digital PLL Output. Programmable from 1MHz to 75MHz. SS2: When Configured as SPI Main, SS2 is a Subordinate Select Output. RO: When Configured as SPI Subordinate, the SPI Main can Read from the Local SPI Subordinate when Initiating an Exchange with the Remote Subordinate. RO is an input.
21	MFP5	ODO5_GPI5* ADC1 LMN0	ODO5_GPI5: Configurable General-Purpose Input or Open-Drain Output. Power-up default is high impedance with a 1MΩ pulldown resistor and GPIO disabled. ADC1: Voltage Monitor Analog Input. LMN0: Line-Fault Monitor Input.

PIN	NAME	FUNCTION MODE	FUNCTION
		GMSL2	
22	MFP6	ODO6_GPI6* ADC2 LMN1	ODO6_GPI6: Configurable General-Purpose Input or Open-Drain Output. Power-up default is high impedance with a 1MΩ pulldown resistor and GPIO disabled. ADC2: Voltage Monitor Analog Input. LMN1: Line-Fault Monitor Input.
31	MFP7	GPI7* GPIO7 MOSI SDA1 RX1 LOCK (Alt)	GPI7: Configurable General-Purpose Input. Power-up default is GPIO with a 1MΩ pulldown resistor, transmit the GPIO value across the GMSL Link. GPIO7: Configurable General-Purpose Input or Output. MOSI: SPI Main Output, Drives Data to an External SPI Subordinate. When configured as the main, push-pull output that drives data to external subordinate. When configured as a subordinate, input with an internal 1MΩ pulldown to ground that receives data from an external main. SDA1: Pass-Through I <sup>2</sup> C Data Input/Output. RX1: Pass-Through UART Input. LOCK (Alternate): High Indicates GMSL Link is Locked with Correct Serial Word Boundary Alignment.
32	MFP8	GPIO8* MS MISO SCL1 TX1 ERRB (Alt)	GPIO8: Configurable General-Purpose Input or Output. Power-up default is pulled high, output driver enabled, ready to receive and output the GPIO value from the GMSL link. MS: Mode Select Input. Selects Base Mode or Bypass Mode for UART Control Channel. MISO: Input, Receives Data from an External SPI Subordinate. SCL1: Pass-Through I <sup>2</sup> C Clock Input/Open-Drain Output with an Internal 40kΩ Pullup to V <sub>DDIO</sub> . TX1: Pass-Through UART Output with an Internal 40kΩ Pullup to V <sub>DDIO</sub> . ERRB (Alternate): Error Output. Goes low when a line-fault on LMN0 or LMN1, or a CSI-2 ECC, CRC, or other (as configured) error is detected.
5	MFP9	SDA_RX* ODO9_GPI9 SDA2_RX2	SDA_RX: I <sup>2</sup> C Serial-Data Input/Output or UART Receive. Internal 40kΩ pullup to V <sub>DDIO</sub> . (SDA or RX selected by CFG0 at power-up) SDA: I <sup>2</sup> C Serial-Data Input/Open-Drain Output. RX: UART Input. ODO9_GPI9: General-Purpose Input and/or Open-Drain Output. SDA2_RX2: Pass-Through I <sup>2</sup> C Serial-Data Input/Output or UART2 Receive. Internal 40kΩ pullup. SDA2: Pass-Through I <sup>2</sup> C Data Input/Open-Drain Output. RX2: Pass-Through UART Input.
6	MFP10	SCL_TX* ODO10_GPI10 SCL2_TX2	SCL_TX: I <sup>2</sup> C Serial-Clock Input/Output or UART Transmit. Internal 40kΩ pullup to V <sub>DDIO</sub> . SCL: I <sup>2</sup> C Clock Input/Open-Drain Output. TX: UART Open-Drain Output. (SCL or TX selected by CFG0 at start-up) ODO10_GPI10: Open-Drain Output or General-Purpose Input. SCL2_TX2: Pass-Through I <sup>2</sup> C Serial-Clock Input/Output or UART2 Transmit. Internal 40kΩ pullup to V <sub>DDIO</sub> . SCL2: Pass-Through I <sup>2</sup> C Clock Input/Open-Drain Output. TX2: UART Pass-Through Open-Drain Output.
<b>POWER SUPPLIES—SEE <a href="#">Table 3</a> FOR DECOUPLING CAPACITOR RECOMMENDATIONS</b>			
11	V <sub>DD18</sub>	V <sub>DD18</sub>	1.8V Analog Supply.
14	CAP_VDD	CAP_VDD	Decoupling Capacitor Pin for 1V Core Supply.
15	V <sub>DD</sub>	V <sub>DD</sub>	Digital Core Supply. Connect a 1.0V to 1.2V supply.
16	V <sub>DDIO</sub>	V <sub>DDIO</sub>	1.8V to 3.3V I/O Supply.

PIN	NAME	FUNCTION MODE	FUNCTION
		GMSL2	
EP	EP	EP	Exposed Pad. EP is the ground connection to the device. EP MUST be connected to the PCB ground plane through an array of vias for proper thermal and electrical performance.
<b>MISCELLANEOUS—SEE <a href="#">Table 3</a> FOR EXTERNAL COMPONENTS</b>			
1	PWDNB	PWDNB	PWDNB: Active-Low Power-Down Input with a $1\text{M}\Omega$ Pulldown to Ground. Set PWDNB low to enter power-down mode.
7	$V_{\text{REF}}$	$V_{\text{REF}}$	Optional External Reference Voltage for ADC. Voltage reference should be $V_{\text{REF}} = 1.25\text{V}$ . Leave open if not used.
8	X1/OSC	X1/OSC	Crystal/Oscillator Input. If a crystal is used, connect to one terminal of a $25\text{MHz} \pm 200\text{ppm}$ crystal and connect a load capacitor from X1/OSC to EP (load capacitor value depends on the crystal used). If an oscillator is used, supply a $25\text{MHz} \pm 200\text{ppm}$ signal. Leave open if using Reference Over Reverse mode.
9	X2	X2	Crystal Input. Connect to one terminal of a $25\text{MHz} \pm 200\text{ppm}$ crystal and connect a load capacitor from X2 to EP (Load capacitor value depends on crystal used). Leave open if using Reference Over Reverse mode or using an oscillator.
10	XRES	XRES	Used to Calibrate SIO Output Driver Swings. Connect a $402\Omega \pm 1\%$ resistor between XRES and Ground (EP).

## Functional Diagrams



## Detailed Description

### Additional Documentation

In addition to the provided information, designers must also use the following information to correctly design systems using the MAX96717.

- *GMSL2 Channel Specification User Guide*
- *GMSL2 Hardware Design Guide*
- *GMSL2 User Guide*
- Device Errata

The Channel Specification contains physical layer requirements for the PCB traces, cables, and connectors that constitute the GMSL link. The Hardware Design Guide contains recommendations for PCB design, applications circuits, selection of external components, and guidelines for use of GMSL signal integrity tools. The User Guide contains detailed programming guidelines for GMSL device features. Errata sheets contain deviations from published device specifications, and are specific to part number and revision ID. Contact the factory for these documents.

### Recommended Operating Conditions, External Component Requirements and ESD Protection

**Table 2. Recommended Operating Conditions**

PARAMETER	PIN	NOMINAL VOLTAGE	MIN	TYP	MAX	UNIT
Supply Range		V <sub>DD18</sub>	1.7	1.8	1.9	V
		V <sub>DD</sub>	0.95		1.26	
		V <sub>DDIO</sub>	1.7		3.6	
Maximum Supply Noise		V <sub>DD18</sub>		25		mV <sub>P-P</sub>
	V <sub>DD</sub>	1.0V		25		
		1.1V		37.5		
		1.2V		50		
	V <sub>DDIO</sub>	1.8V		50		
		3.3V		100		
Operating Junction Temperature, T <sub>J</sub>			-40		+125	°C

**Note:** Supply noise < 1MHz. Supply voltage ripple is assumed to be symmetric around the measured DC supply voltage. For example, 50mV<sub>P-P</sub> means  $\pm 25\text{mV}$  peak voltage.

See [Figure 20](#), [Figure 21](#), and the [Typical Application Circuits](#) section for typical application circuits. [Table 3](#) details critical components that must be connected to the specified pins for correct functionality.

**Table 3. External Component Requirements**

COMPONENT	SYMBOL	CONDITION	VALUE	UNIT
XRES	R <sub>XRES</sub>	Connect R <sub>XRES</sub> resistor between XRES pin and ground. Total variation not to exceed $\pm 3\%$ including tolerance, temperature, and lifetime drift (e.g. $\pm 100\text{ppm}/^\circ\text{C}$ temperature coefficient and $\pm 1\%$ lifetime drift).	402 $\pm 1\%$ . Use a single resistor.	Ω
Line-Fault Pulldown Resistor	R <sub>PD</sub>	Connect to ground at far end of Coax/STP cable; only required if line-fault detection is used. Total variation not to exceed $\pm 3\%$ including tolerance, temperature, and lifetime drift (e.g. $\pm 100\text{ppm}/^\circ\text{C}$ temperature coefficient and $\pm 1\%$ lifetime drift).	49.9 $\pm 1\%$	kΩ

**Table 3. External Component Requirements (continued)**

COMPONENT	SYMBOL	CONDITION	VALUE	UNIT
Line-Fault Resistor	R <sub>EXT</sub>	Connect to LMN input of serializer at near end of Coax/STP cable; only required if line-fault detection is used. Total variation not to exceed $\pm 3\%$ including tolerance, temperature and lifetime drift (e.g. $\pm 100\text{ppm}/^\circ\text{C}$ temperature coefficient and $\pm 1\%$ lifetime drift). Line-fault detection cannot be used in conjunction with PoC.	LMN0 STP mode LMN0 Coax mode LMN1	$42.2 \pm 1\%$ $48.7 \pm 1\%$ $48.7 \pm 1\%$
Link-Isolation Capacitors	C <sub>LINK</sub>	Place in series and in close proximity to the SIO pins (pins 12 and 13).	0.1	$\mu\text{F}$
Termination Resistors for SIO_N pins in Coax mode	R <sub>TERM</sub>	Connect in series with C <sub>LINK</sub> capacitor between SION and ground when GMSL link is configured in Coax mode. Place near associated SION pin.	49.9 $\pm 1\%$	$\Omega$
Crystal		Place as close as possible to pins X1/OSC (pin 8) and X2 (pin 9), and connect between these two pins.	25MHz $\pm 200\text{ppm}$	
Crystal Load Capacitors		Use crystal-loading capacitor guidance from the crystal manufacturer. Select values that compensate for the X1 and X2 input, and PCB node capacitances. Place the capacitors as close as possible to pins X1/OSC (pin 8) and X2 (pin 9).	Crystal dependent	pF
V <sub>DDIO</sub> Decoupling Capacitors*		Place 0.01 $\mu\text{F}$ , 0.1 $\mu\text{F}$ capacitors as close as possible to pin V <sub>DDIO</sub> (pin 16). Include a minimum of 10 $\mu\text{F}$ bulk decoupling on the PCB.	0.01 + 0.1 + 10	$\mu\text{F}$
V <sub>DD18</sub> Decoupling Capacitors*		Place 0.01 $\mu\text{F}$ , 0.1 $\mu\text{F}$ capacitors as close as possible to pin V <sub>DD18</sub> (pin 11). Include a minimum of 10 $\mu\text{F}$ bulk decoupling on the PCB.	0.01 + 0.1 + 10	$\mu\text{F}$
V <sub>DD</sub> Decoupling Capacitors*		Place a 0.01 $\mu\text{F}$ , 0.1 $\mu\text{F}$ capacitors as close as possible to pin V <sub>DD</sub> (pin 15). Include a minimum of 10 $\mu\text{F}$ bulk decoupling on the PCB. See configuration information in the <a href="#">Power Supplies</a> section.	0.01 + 0.1 + 10	$\mu\text{F}$
CAP_VDD Decoupling Capacitor		Place a 0.1 $\mu\text{F}$ capacitor as close as possible to pin 14. Include a minimum of 10 $\mu\text{F}$ bulk decoupling near the pin.	0.1 + 10	$\mu\text{F}$
Open-Drain Pullup Resistors		Application-specific. Quantity and values depend on multifunction GPIO pin configurations.		
Resistors for Configuration Pin Resistor Divider	R1, R2	Place resistor-divider close to pin 3 (MFP1/CFG0).	Use $\pm 1\%$ Tolerance Resistors. See <a href="#">Table 11</a> .	$\Omega$
	R1, R2	Place resistor-divider close to pin 4 (MFP2/CFG1).	Use $\pm 1\%$ Tolerance Resistors. See <a href="#">Table 12</a> .	$\Omega$

\* With exception of CAP\_VDD, power supply decoupling capacitor values are recommendations only. It is the responsibility of the board designer to determine what decoupling is necessary for the specific application.

**Table 4. ESD Protection**

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNIT
SIO_	V <sub>ESD</sub>	Human Body Model (HBM), R <sub>D</sub> = 1.5k $\Omega$ , C <sub>S</sub> = 100pF		$\pm 8$		kV
		ISO 10605, R <sub>D</sub> = 330 $\Omega$ , C <sub>S</sub> = 150pF, Contact Discharge		$\pm 6$		

**Table 4. ESD Protection (continued)**

		ISO 10605, $R_D = 330\Omega$ , $C_S = 150\text{pF}$ , Air Discharge	$\pm 8$		
		AEC-Q100-011 Rev-C1, Charged Device Model (CDM)	750		V
All Other Pins	$V_{ESD}$	Human Body Model/HBM, $R_D = 1.5\text{k}\Omega$ , $C_S = 100\text{pF}$	$\pm 4$		kV
		AEC-Q100-011 Rev-C1, Charged Device Model/CDM	750		V

## Functional Diagrams

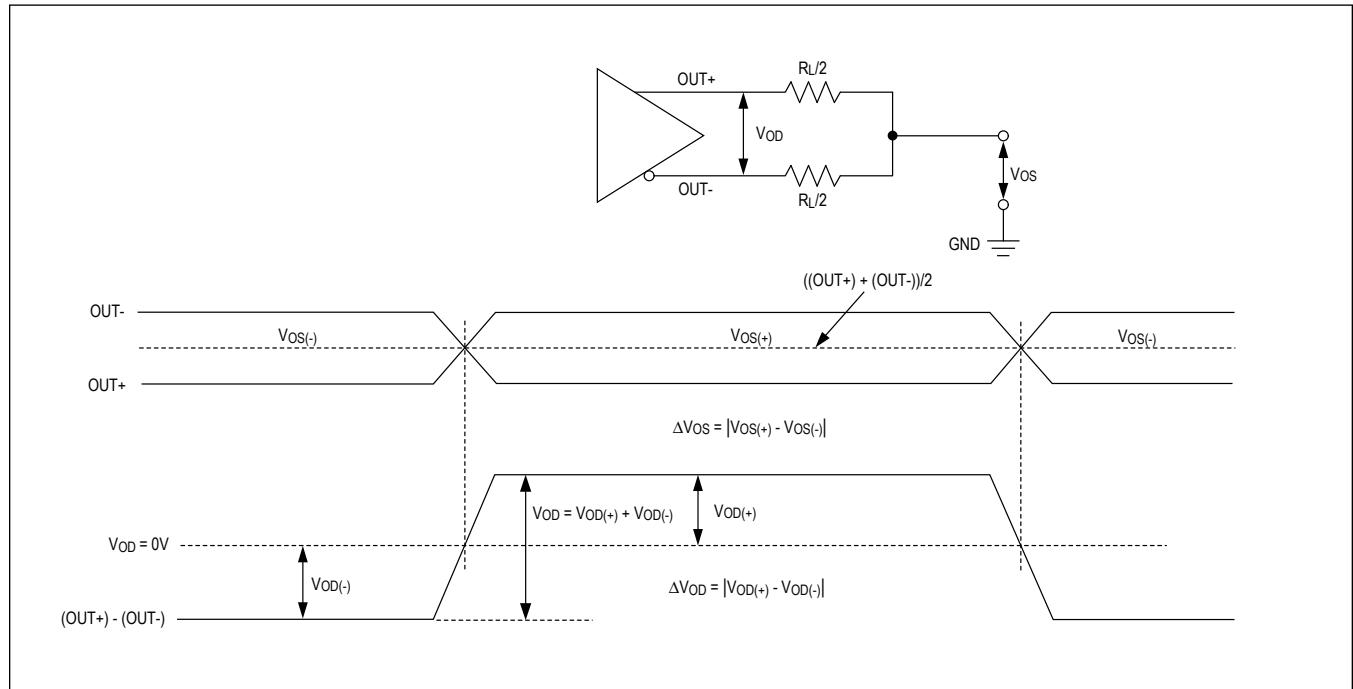


Figure 1. Serial Output Parameters

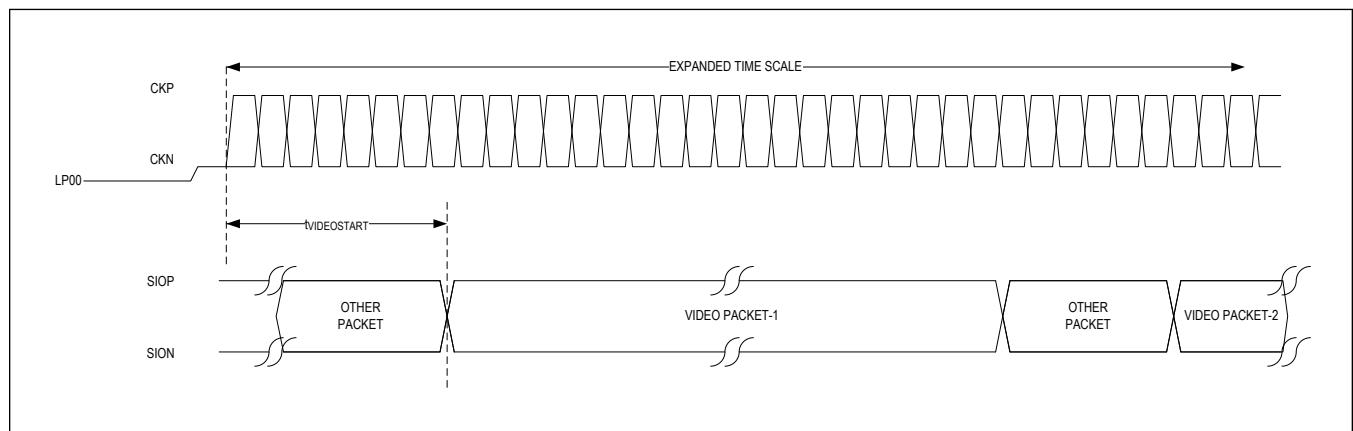


Figure 2. GMSL2 Data Initialization Time

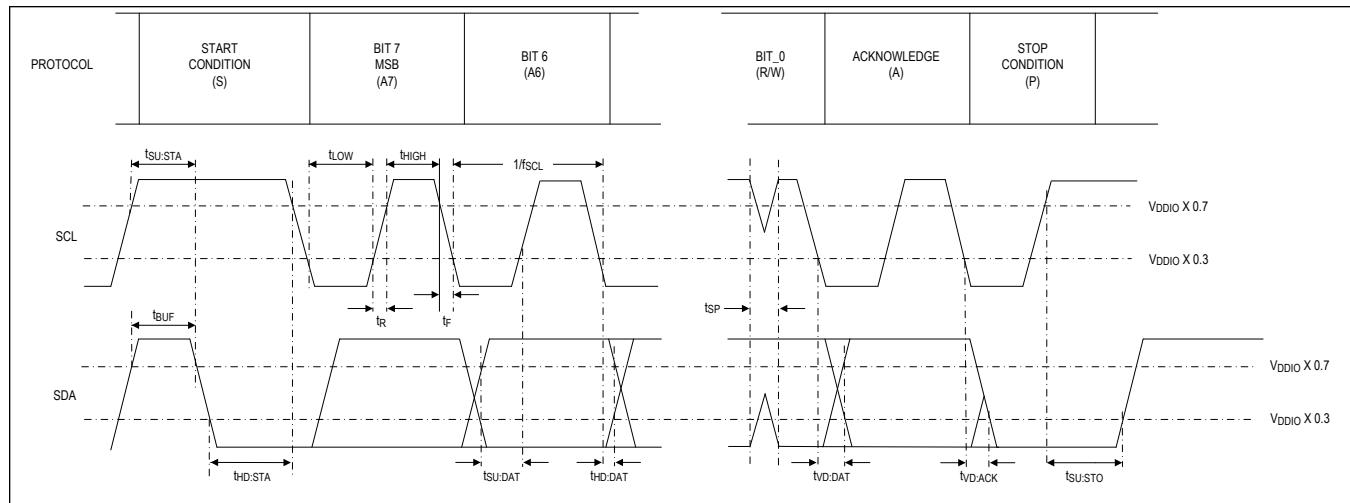
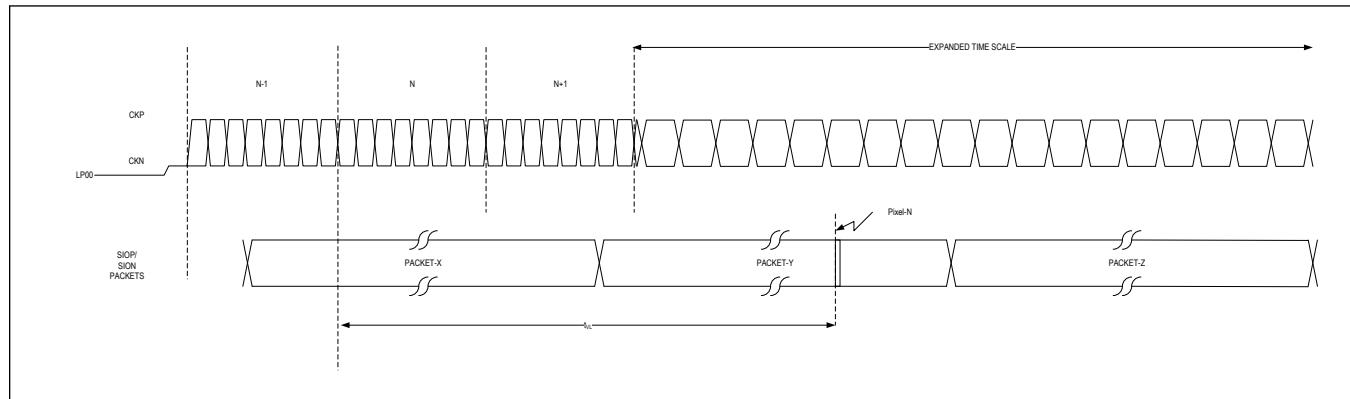
Figure 3.  $I^2C$  Timing Parameters

Figure 4. GMSL2 Data Latency

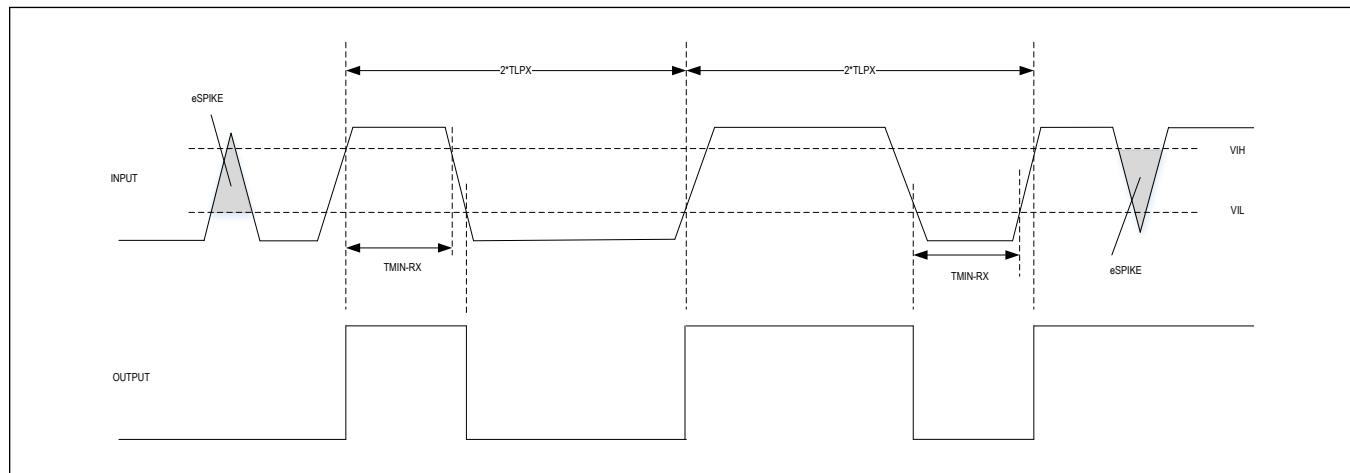


Figure 5. D-PHY LP Receiver Pulse

Copyright © 2007-2016 MIPI Alliance, Inc. All right reserved.

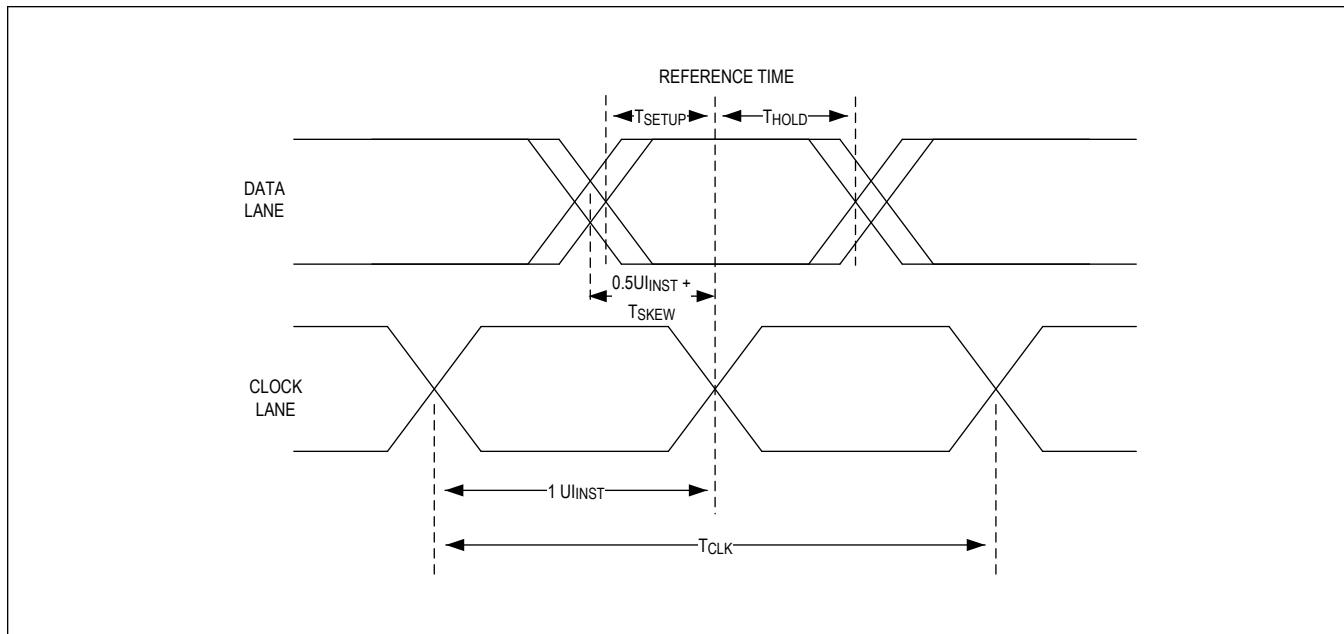


Figure 6. D-PHY Data Clock Timing

Copyright © 2007-2016 MIPI Alliance, Inc. All rights reserved.

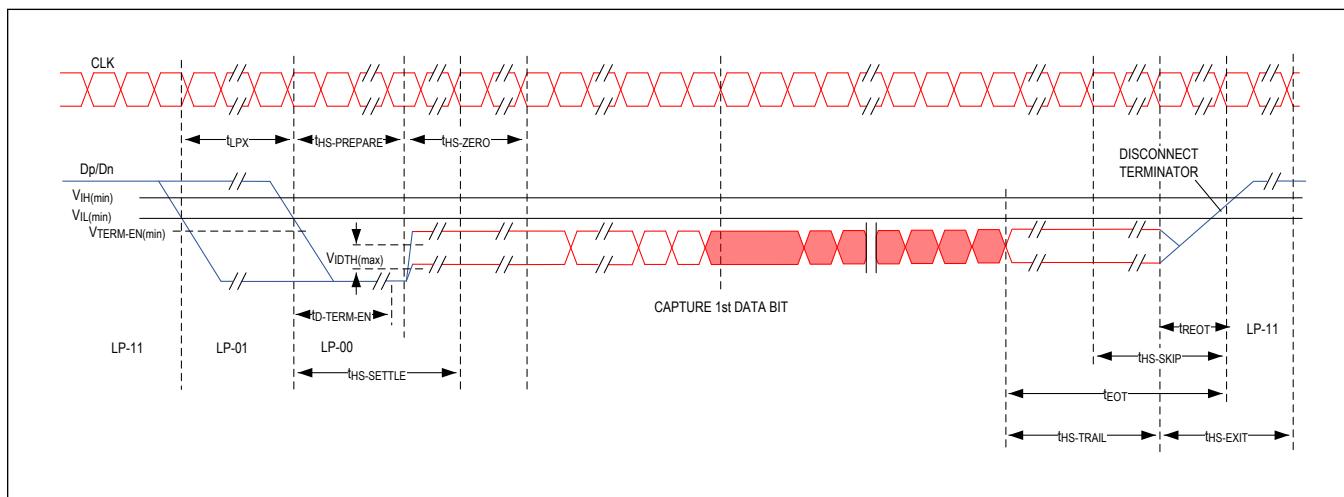


Figure 7. High-Speed Data Transmission in Bursts

Copyright © 2007-2016 MIPI Alliance, Inc. All rights reserved.

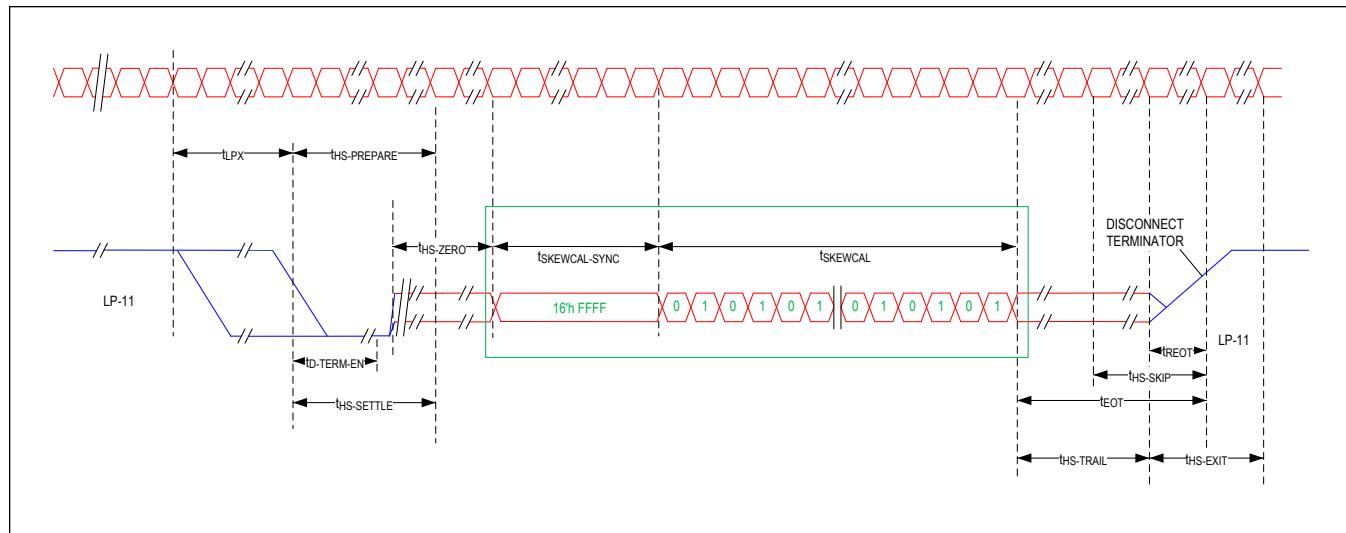


Figure 8. D-PHY High-Speed Skew Calibration

Copyright © 2007-2016 MIPI Alliance, Inc. All rights reserved.

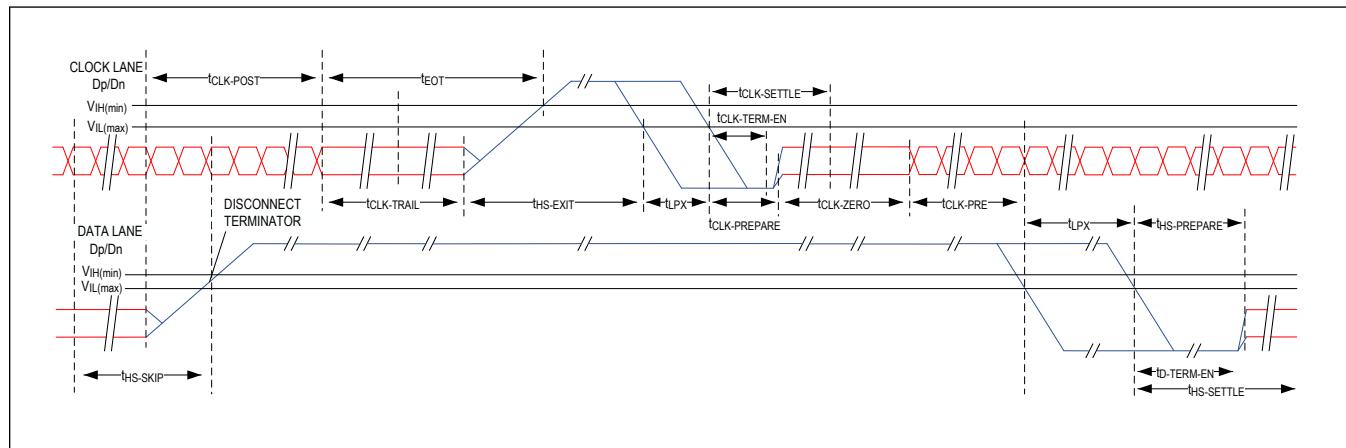


Figure 9. Switching the Clock Lane Between Clock Transmission and Low-Power Mode

Copyright © 2007-2016 MIPI Alliance, Inc. All rights reserved.

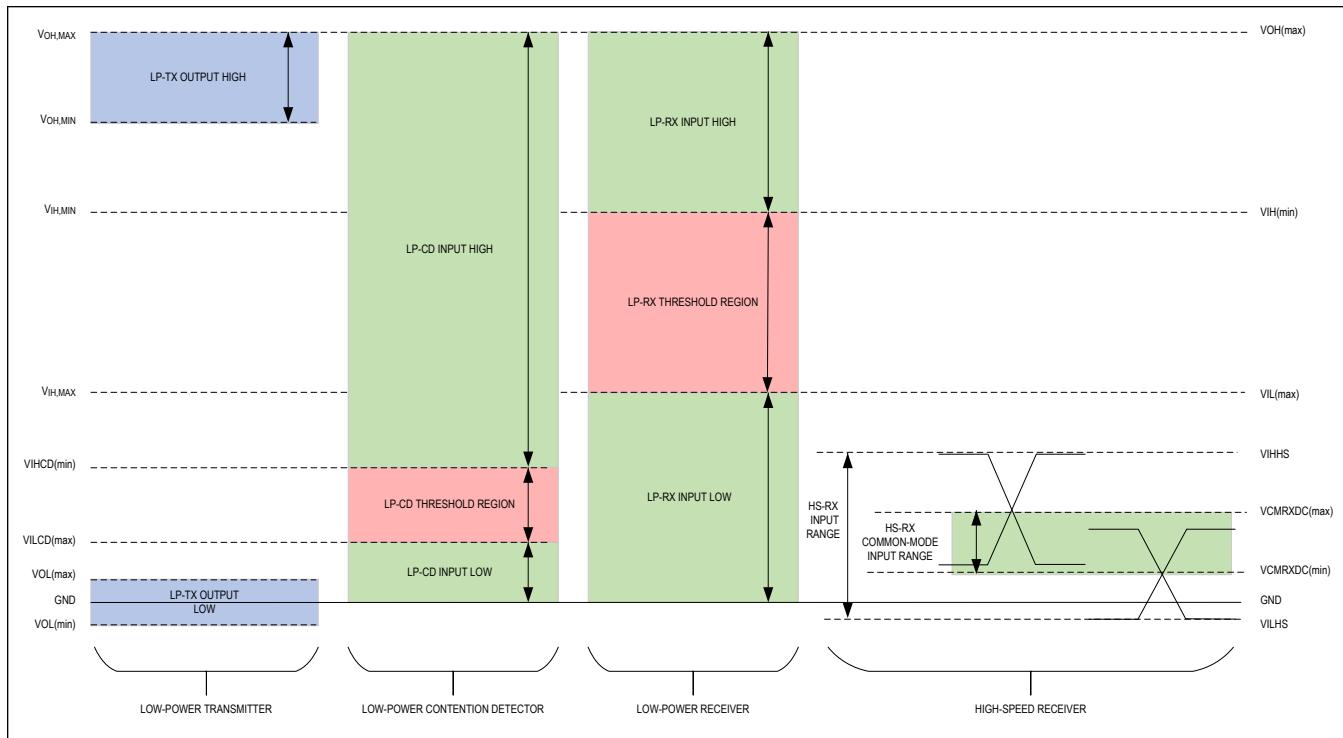


Figure 10. Signaling and Contention Voltage Levels

Copyright © 2007-2016 MIPI Alliance, Inc. All rights reserved.

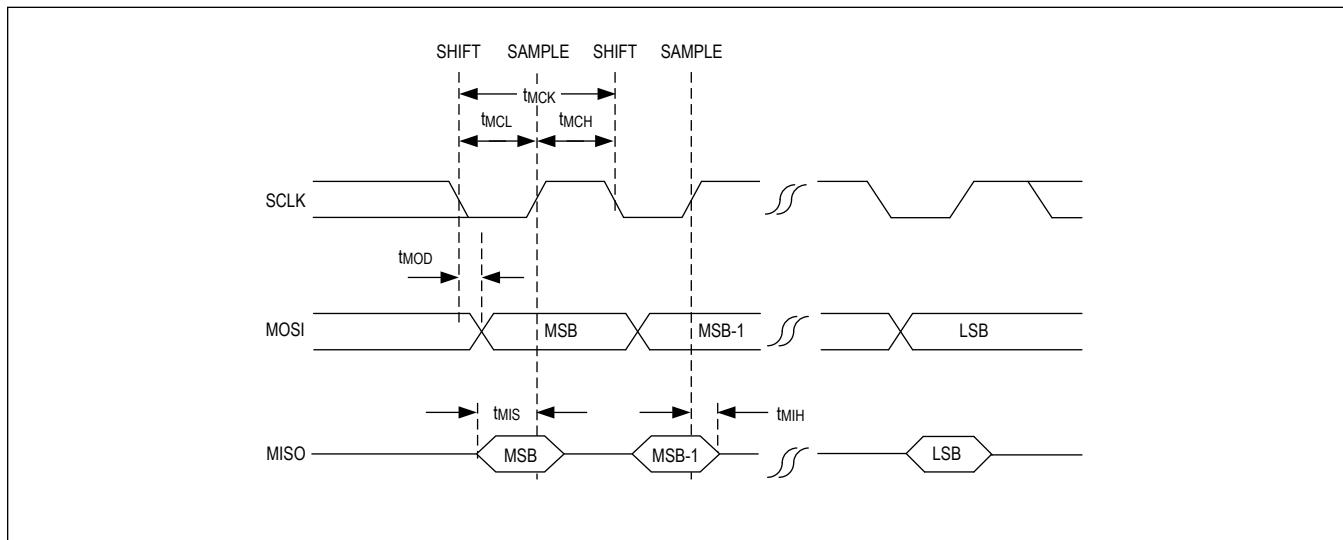


Figure 11. SPI Main-Mode Timing Parameters

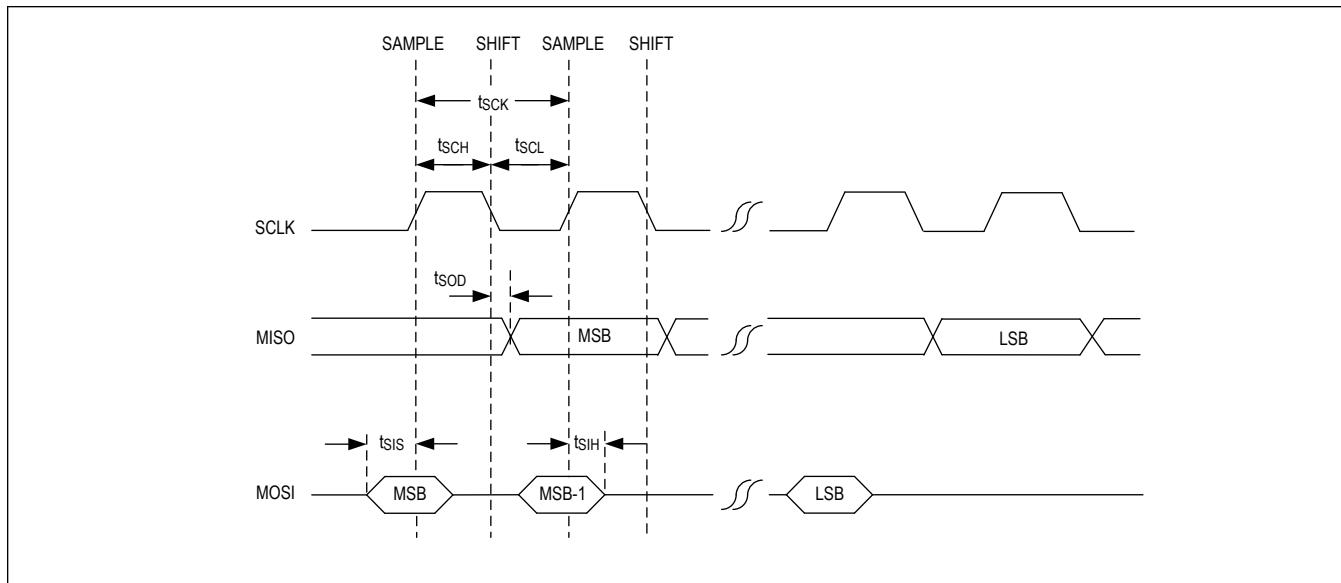


Figure 12. SPI Subordinate-Mode Timing Parameters

## Introduction

Analog Devices' tunneling GMSL2 serializers and deserializers provide end-to-end data integrity of both sensor data and side-channel data, while also providing sophisticated link management for high-speed, low bit-error-rate, and serial data transport. They support a comprehensive suite of sensor and communication interfaces over a single wire. Tunneling GMSL2 serializers provide up to 6Gbps forward and 187.5Mbps reverse packetized data transmission over each fixed-speed link.

The following sections provide a brief overview of the device functions and features. Contact the factory for additional information, and details on configuration of each function and feature.

## Product Overview

The MAX96717 serializer converts MIPI CSI-2 to single-link GMSL2. See [Figure 13](#). It also sends and receives side-channel data, enabling full-duplex transmission of forward-path sensor data and bidirectional data over low-cost 50Ω coax or 100Ω STP cables meeting the GMSL2 channel specification.

The MAX96717 has a four-lane D-PHY v1.2 that supports a data rate of 108Mbps to 2.5Gbps per lane. The number of active D-PHY v1.2 data lanes is programmable as one, two, three, or four lanes. Up to 16 virtual channels are also supported.

The MAX96717 is intended for use with a GMSL2 deserializer such as the dual-port MAX96716A/B, a quad-port MAX96712, or a GMSL3 deserializer operating in GMSL2 mode. For example, when used with the MAX96716A, several modes are supported. A single sensor can be connected to an SoC in GMSL2 mode at a data rate of 3Gbps or 6Gbps, as shown in [Figure 13](#). Two MAX96717 serializers can be used with one MAX96716A/B deserializer to connect two sensors to two SoC devices. See [Figure 14](#). The cable types, sensor timing, and data rates do not have to be the same. The links also do not need to be set to the same forward rate (3Gbps or 6Gbps). Data from Link A and Link B are output on separate, dedicated CSI-2 ports for capture by the SoCs.

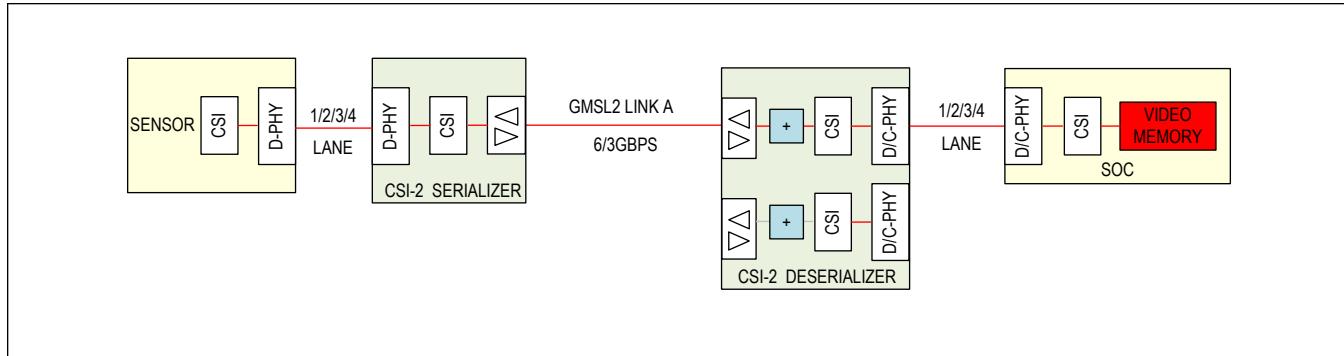


Figure 13. Single Link

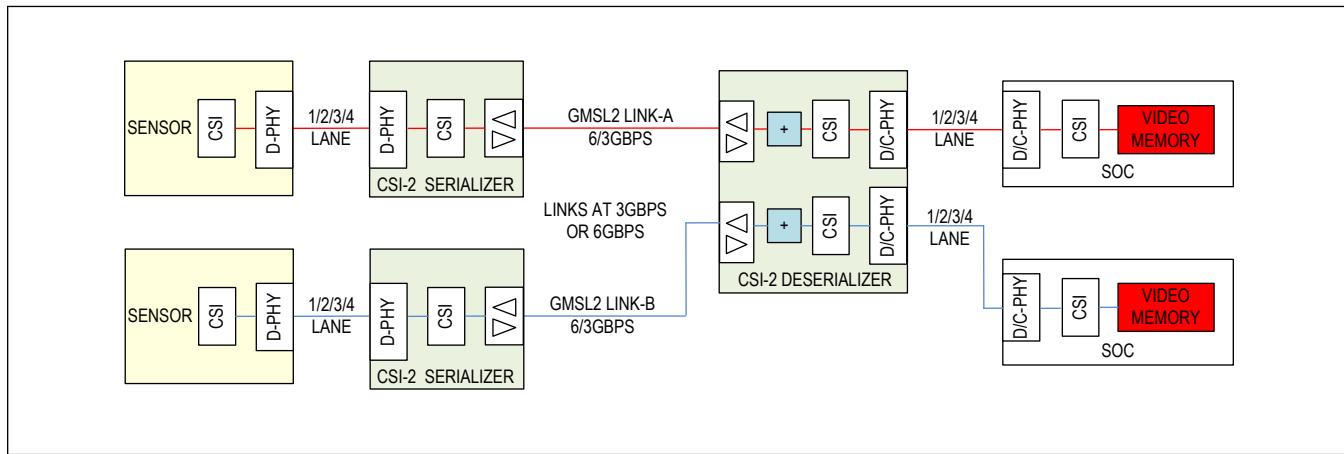


Figure 14. Dual Link, Separate Data Type

The MAX96717 supports data aggregation, as detailed in [Figure 15](#). The sources can have different video timing and resolution, and the serial links can be set to different forward rates (3Gbps or 6Gbps). The SoC identifies the video source by reading the packet's virtual channel. If both sources use the same virtual channel, the MAX96717 can assign a different virtual channel (16 are available) in Pixel mode. Reassignment of up to 16 data types is also possible in Pixel mode. The aggregated data can be replicated to both MIPI ports, allowing multiple SoCs to process the same data as shown in [Figure 16](#).

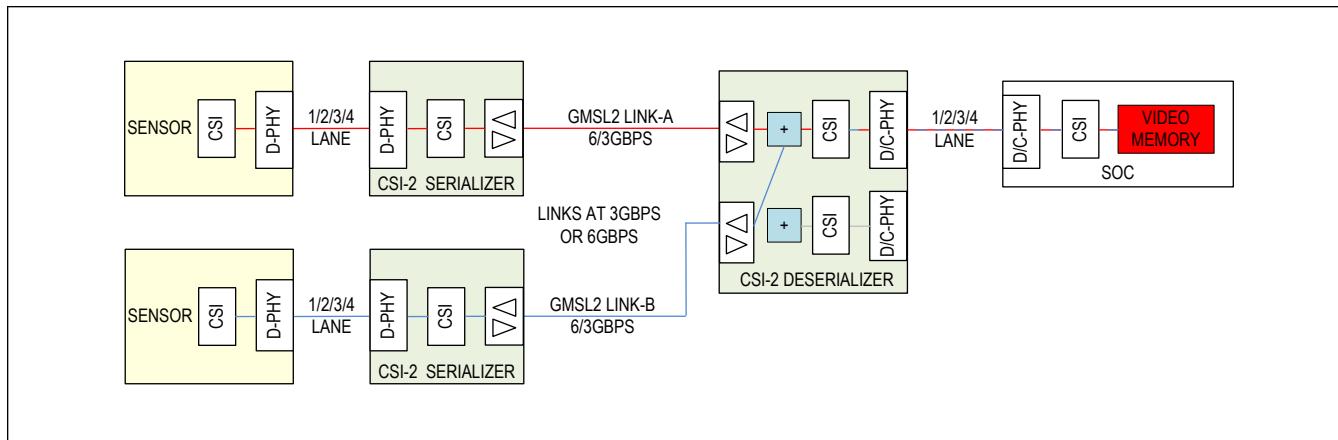


Figure 15. Dual Link, Data Aggregation

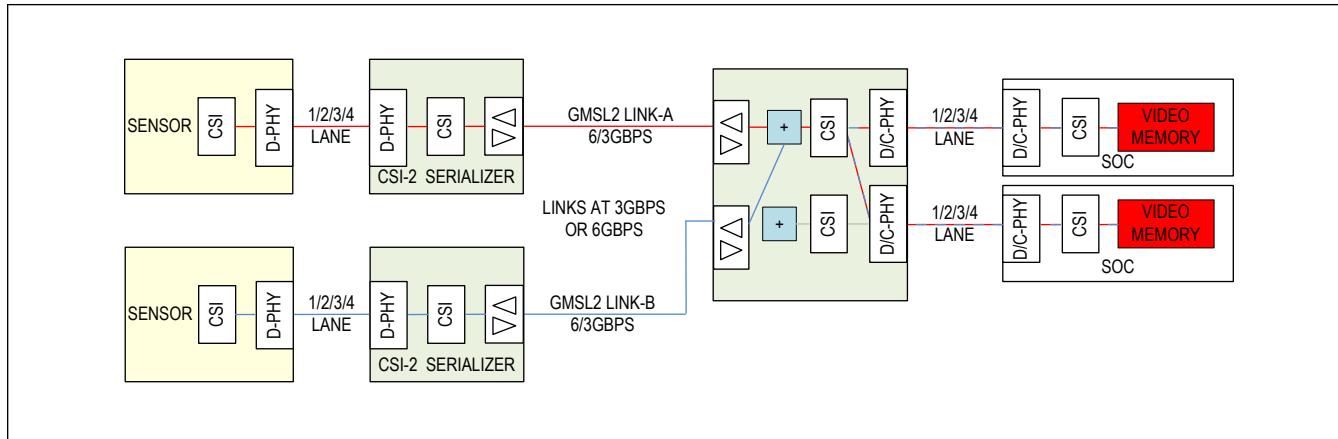


Figure 16. Dual Link, Data Aggregation and Replication

The MAX96717 is an ideal sensor serializer to use (it has a CSI-2 interface and single or multiple data types) when various cameras and/or sensors need to be supported. Radio detection and ranging (RADAR) and light detection and ranging (LiDAR) sensors can also be supported where a high-speed serial peripheral interface (SPI) is generally required.

### Tunneling and Pixel Modes

The MAX96717 is specifically designed for advanced driver assistance systems (ADAS), where data integrity is a key safety requirement. Prior GMSL2 solutions supported only Pixel mode for transporting received data from a MIPI CSI-2 interface over the GMSL link. In Pixel mode, the CSI-2 data is depacketized at the serializer's CSI-2 input interface. The received CSI-2 packet header includes an error correction code (ECC), which is checked and removed at the serializer input. The received CSI-2 packet footer contains the CSI-2 cyclic redundancy check (CRC), which is also checked and removed.

Video line pixel data and video routing information, such as data type and virtual channel, are received and extracted at the CSI-2 interface. Both video pixel data, control channel data, and routing information are input into a scheduler in the serializer. The scheduler packetizes and encapsulates the data using GMSL protocol and sequences data transmission across the GMSL link. Video data transport across the GMSL link is protected by line CRCs that are part of the GMSL protocol.

The deserializer receives the GMSL packets and verifies the GMSL2 line CRCs. A CSI-2 interface at the deserializer output encapsulates each video line using CSI-2 protocol and outputs it in CSI-2 format across a CSI-2 interface to the SoC (see [Figure 17](#)).

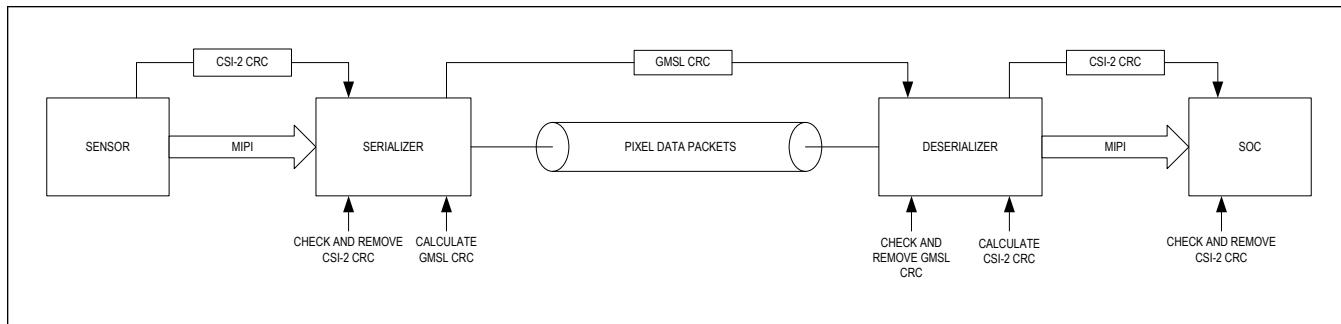


Figure 17. Pixel Mode

In Tunneling mode, the received CSI-2 ECC byte and CRC bytes are checked at the serializer input. These, as well as routing and pixel data, are received as a byte stream. The byte stream is split into smaller packets that are encapsulated using GMSL2 protocol.

The serializer adds a line CRC, protecting transmission across the GMSL channel. This CRC covers the entire GMSL2 packetized byte stream for a video line (see [Figure 18](#)). The deserializer receives the transmitted GMSL2 packets and control channel packets, checks and removes the GMSL CRC, separates the video data from control data, and reconstructs each received CSI-2 packet that is output to the SoC on a CSI-2 interface. A CRC is calculated on the video data output on the CSI-2 interface. This CRC is compared by the deserializer to the original CRC received from the video source. This comparison guarantees that the entire data packet output on the standard MIPI interface is identical to that received at the serializer input. Tunneling mode is more bandwidth-efficient if multiple data types are being sent. Because data received at the serializer input and data output from the deserializer are verified to be identical, Tunneling mode does not allow for the processing of video data, such as watermarking or lossy data compression. Different data rates and lane counts on the serializer and deserializer are still possible.

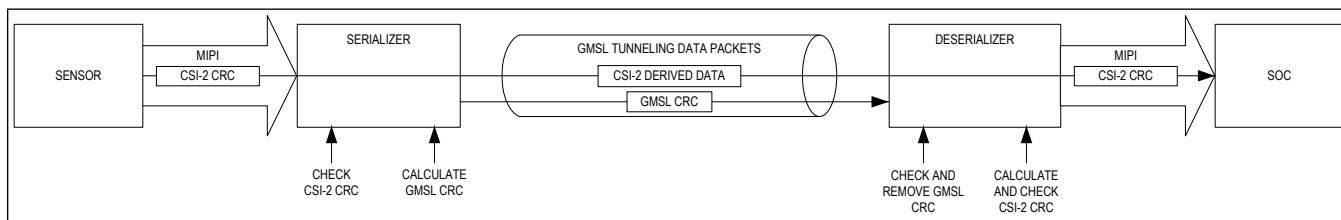


Figure 18. Tunneling Mode

## Video Pipeline

The video channel is designed to transmit video data received from the CSI-2 interface to the deserializer side of the link. The following data types are supported: RGB888, RGB666, RGB565, YUV422-8, YUV422-10, RAW8, RAW10, RAW12, RAW14, RAW16, RAW20, User-Defined, Embedded, and Null. By default, the bits per pixel (bpp) of the video stream is automatically detected by inspecting the data type of the CSI-2 packets. If a video stream with a different bpp is sent through the same video pipe in Pixel mode, disable the auto-detect through a register write. Video input data consists of color, HS, and VS synchronous to the pixel clock (PCLK). Reception of video is detected using the PCLKDET function. Video flows through the design as described in the following sections.

### Video Pipes in Pixel Mode

Video data transport in GMSL2 Pixel mode is based on the concept of video pipes. Carrying data in pipes allows GMSL2 to bridge different digital video interfaces and perform watermark generation and detection (Note: Watermark generation is not supported by this device.). A pipe carries video stream(s) and video synchronization data and operates in one of the following modes:

- Mode 1: Streams with constant bits per pixel (bpp) of up to 24bpp. The bpp of the streams must be the same.
- Mode 2: Streams with 16, 14, 12, 10, or 8bpp. Streams less than 16bpp are padded with zeros.

- Mode 3: Streams with two different bpp rates. The bpp of one stream must be twice the bpp of the other stream. The highest bpp stream is 24bpp.

In all modes, a pipe can carry multiple concurrent video streams, with each stream having different virtual channels and data types.

Modes 1 and 3 carry data at full bandwidth, but put more restrictions on bpp than Mode 2. Mode 2 allows streams with different bpp rates, but streams of less than 16bpp are carried using more bandwidth than necessary on the GMSL2 link because of zero-padding. Modes 1 or 3 are sufficient for most applications. Mode 2 requires less programming and is more convenient if the application does not require maximum link bandwidth.

The MAX96717 has a single input port and one video pipe. The pipe in the MAX96717 has a dedicated MIPI receiver buffer. The retiming buffer has the capacity to buffer 96 24-bit pixels. This allows sufficient memory for transmission of a line to start and not overflow or underflow. The number of pipes used by a deserializer is equal to the number of pipes used by all connected serializers.

After data exits a retiming buffer, it goes through a crosspoint switch, and a data type (DT) and virtual channel (VC) reassignment stage. If the video source has a CSI-2 output, packet, DT and VC can each be left as-is or reassigned by register programming. Up to 16 DT/VC incoming pairs can be mapped to 16 DT/VC outgoing pairs.

### RGB888 Video-Pattern Generator

The RGB888 video-pattern generator (VPG) is an auxiliary block that can be used for a variety of purposes, such as replacing video from the peripheral with the video pattern generated by the VPG. Various patterns can be generated when configured using the configuration registers. The VPG creates a checker board pattern and gradient using RGB888 data.

**Note:** The VPG is only for use in Pixel mode.

### Tx Crossbar

Data from the video input is captured and optionally multiplexed with the generated patterns from the VPG. Final data goes into a crossbar multiplexer that allows any input bit to be mapped to any output bit using the configuration registers. The crossbar should only be used in Pixel mode.

### GMSL2 Packet Protocol

#### Packet Protocol

GMSL2 is a fixed rate, packet-based transmission protocol designed to efficiently and dynamically carry multiple types of communication channels concurrently. Link bandwidth is only used by a channel when data is being sent. The link bit rate is based on a constant frequency link clock generated from the 25MHz crystal oscillator, external reference frequency, or RoR feature. The link clock does not have any relation to the video-pixel clock.

GMSL2 uses a packet-based protocol to seamlessly share the link bandwidth between communication channels in a flexible way. Bandwidth allocation is dynamic, so that if a certain channel is not active, it does not consume any link bandwidth, and all the remaining active channels can share the full link bandwidth. Maximum packet size is limited to prevent a single channel from utilizing the link bandwidth for an extended time. In most cases, available link bandwidth exceeds the bandwidth requirement. Idle packets are used to fill in the unused link bandwidth.

The same data protocol is used on forward and reverse channels, and for both video and control-channel data.

#### 9b10b Link Encoding

9b10b is a link encoding scheme that maps 9-bit symbols (512 unique data symbols) and eight additional special characters (K) to 10-bit symbols. This is an efficient encoding scheme that requires significantly less encoding overhead than other similarly robust coding schemes. To illustrate, this method requires 11% overhead, whereas 8b10b encoding requires 25% overhead – effectively apportioning more bandwidth for communications and data transfer. The link is DC-balanced: the running disparity (i.e., the number of 1s and 0s in a given window) is bounded to  $\pm 9$  with a maximum run length (i.e., consecutive number of 1s or 0s) limited to seven.

Eight available special K characters are used for link synchronization, as packet delimiters and for other control purposes. The Hamming distance (number of bit differences) in this scheme is at least 3 between any two special symbols.

Therefore, a special symbol cannot transform into another special symbol with one or two bit errors. A cascading sequence of special symbols creates a unique bit pattern that does not exist in any other data symbol combination; this sequence is deployed for symbol and word boundary locking purposes to delimit data packets.

Each packet starts with two words consisting of two special symbols in each word. This packet indicator is repeated for error-correction purposes and allows one bit-error correction in the receiver for errors that hit packet start words.

**Note:** *Reduction of packet start words to one word (two special symbols) to increase bandwidth efficiency is a programmable option. This option should only be employed if the physical link is sufficiently reliable and the additional usable bandwidth is needed. In this scenario, a bit error hitting one of the special symbols in the packet start word does not transform the affected packet into another packet; rather, the packet is dropped entirely. Dropped control packets can be automatically recovered by the Automatic Repeat Mechanism (ARQ) (see the following). Packets can be protected by an optional 16-bit packet Cyclic Redundancy Check (CRC) (see the following).*

### Tunneling Mode

Tunneling mode transmits all MIPI CSI-2 content, without modification, over the serial link. Unlike Pixel mode, the received CSI-2 payload is not decoded. Unchanged CSI-2 bytes are split into smaller GMSL packets that are encapsulated and protected with GMSL CRCs, and are transported across the link. This means the CSI-2 headers and footers generated by the video source, including ECC and CRC, are preserved through the GMSL link, and can be decoded by the host connected to the deserializer. This provides end-to-end data integrity critical for ADAS vehicles.

Pixel mode checks the ECC and CRC packets for errors at the serializer, and then removes them prior to converting the payload to pixels and transmitting over the GMSL2 link. In Pixel mode, the deserializer recalculates and inserts the CSI-2 ECC and CRC packets. Both Tunneling and Pixel modes have CRC protection of the serial-link payload so any GMSL link errors are detected. However, an error in recalculating the ECC/CRC packets in the deserializer is not detected by the SerDes, but is detected by the MIPI video sink device.

### CRC Generation and Checking

Every packet (excluding Idle and Acknowledge packets) can be protected by a 16-bit packet CRC. Each packet type can be individually configured to enable or disable packet CRC. By default, all low-bandwidth channels have packet CRC. Note that since Acknowledge packets contain no data and the header is repeated, CRC is unnecessary. Although the video channel disables packet CRC in default settings in order to maximize usable bandwidth, video-line CRC (a 32-bit code at the end of each DE or HS pulse) is enabled by default to provide data protection.

The 16-bit packet CRC generator polynomial is:  $x^{16} + x^{15} + x^2 + 1$ .

## Control-Channel Retransmission on Error

### Automatic Repeat Request/Automatic Retransmission (ARQ)

Communications channels with control data (I<sup>2</sup>C/UART, GPIO, SPI) are relatively low bandwidth, but require the highest data integrity protection. An optional automatic packet retransmission method, Automatic Repeat Request (ARQ), is employed here. ARQ works in conjunction with 16-bit packet CRC to detect if packets are received with or without error.

Packets are appended with a 2-bit sequence number on the transmit side, and an acknowledge is sent from the receiver side upon successful receipt of each data packet. These packets are stored on the transmit side until acknowledged. If the acknowledge does not arrive in a predetermined interval or the sequence number of the acknowledge does not match the expected value, the packet waiting at the top of the queue is automatically retransmitted.

The acknowledge packet uses the same header field as low bandwidth packets, but 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, the header symbol is sent twice in the packet and checked against each other to ensure a match. The acknowledge packets also include a 2-bit sequence number that is the same sequence number of the correctly received data packet. The data packet transmitter keeps track of which packets are acknowledged.

## Forward-Error Correction

The MAX96717 includes Forward-Error Correction (FEC), which is optional for GMSL2 (6Gbps or 3Gbps). The Reed Solomon encoding adds a 6-bit correction word to every 121 bits of data for an effective throughput of 93.3% in return for a bit-error-rate (BER) reduction from 1e<sup>-15</sup> to 1e<sup>-55</sup>. Note that use of FEC in the MAX96717 serializer requires pairing it with a deserializer that also supports FEC.

## Scheduler/Arbiter

A scheduler transmits packets with high-priority values before lower priority values. Each communication transmit adapter sets a priority for the packet request before transmitting data to the remote side. The priority value is 2 bits, allowing for four different settings: 0 = Low, 1 = Normal, 2 = High, and 3 = Urgent. The scheduler, provided there is sufficient link bandwidth, chooses to transmit the packet with the highest priority among the pending active requests. Priority levels become more important as link bandwidth becomes scarce. Prioritization should be assigned accordingly.

In most cases, each transmitter adapter should use the normal priority setting (priority = 1) to allow the scheduler to choose the transmission schedule based on recent bandwidth usage. Packet priority can be increased if packets require low latency or have waited for a length of time. For example, packets with maximum latency requirements can have increased priority if the packet request has not been serviced until half of the maximum latency requirement has elapsed. This can also be used for communications channels with continuous data flow (e.g., video). Priority is increased when the transmit adapter data buffer approaches overflow. Conversely, register configuration allows overriding the priority settings of each channel. This option is useful if the host µC wants to prioritize one channel over the others.

**Note:** Communications channels with very relaxed latency requirements can use the low-priority setting (priority = 0). These low-priority packets are not serviced by the scheduler if there are any pending requests of a higher priority setting. In this arrangement, link bandwidth assignment can reach the theoretical maximum for video. Low-priority packets can then be transmitted during video horizontal blanking time, during which the video channel bandwidth usage drops considerably.

## Bandwidth Sharing

The GMSL2 link bandwidth is flexible to share between different communication channels, including video/data, I<sup>2</sup>C/UART control channel, pass-through I<sup>2</sup>C/UARTs, SPI, and GPIOs, as well as various protocol-specific data exchanges, such as information frames, sync, and acknowledgments. Each channel must request the link for packet transmissions. This flexibility comes from packet-based transmission format and dynamic bandwidth allocation. If a certain channel is not active, it does not consume any link bandwidth, leaving the full link bandwidth available for all active communication channels to share. The packet-based protocol allows the sharing requirement to be fulfilled. The maximum packet size is limited to 64 20-bit words to prevent one single channel from monopolizing the link bandwidth and to ensure other channels are served. The total-link bandwidth used by all communication channels cannot exceed the fixed available link bandwidth or errors occur.

The data and control-channel packets can be assigned a priority level. There are four priority levels: low, normal, high, and urgent. The scheduler transmits the packet with the highest priority among the pending requests. Packets with maximum latency requirements can be assigned an increased priority.

A bandwidth sharing scheme is employed in cases of multiple pending requests of the same priority setting to avoid creating buffer overflows. Without bandwidth sharing, a burst data request from a channel could cause buffer overflow in other communications channels on the link. Bandwidth sharing, however, considers predefined bandwidth share ratios and recent bandwidth usage averages of each type of communication channel to avoid buffer overflows, and ensure all channels are served.

The arbiter continuously measures the bandwidth usage of each channel and filters the data according to the analysis of the moving averages. This data is then compared to assigned bandwidth share ratios. To ensure link bandwidth parity, the arbiter takes this recent bandwidth usage information to decide which channel is allocated to use the link for each new packet transmission request. Consider a link allocated for either video or SPI communications, where video is assigned 90% link share and SPI is given 5%. A burst SPI transmission request on this link could result in video overflow buffering if the SPI bandwidth share exceeds 10% of the total. To avoid video buffering, SPI is allotted no more than 10% of the link to preserve the remaining 90% for video transmission.

## GMSL2 Physical Interface

### Link Speeds (Forward and Reverse)

Analog Devices' GMSL2 family of serial links have transmitter and receiver capability enabled simultaneously, enabling full-duplex operation on a single wire. A single cable between the serializer and deserializer delivers data being transmitted from each end of the link. Forward transmission is data being sent from the serializer to the deserializer at a link rate of 3Gbps or 6Gbps. Reverse transmission is data being sent from the deserializer to the serializer at a link rate of 187.5Mbps. The GMSL2 link rates are fixed and independent of the video clock.

### Cabling Options

GMSL2 supports either 50Ω Coax or 100Ω Shielded-Twisted Pair (STP) cabling. Cable attenuation and return loss characteristics must stay within the requirements of the GMSL2 channel specification to achieve robust full-duplex link performance. These requirements vary with selected link rate. The available link rates and GMSL2 adaptive equalization enable support of a wide range of cabling options.

Coax or STP mode, and data rates are configured upon start-up by the level of CFG1 at power-up, and determine which cabling option applies. See [Table 12](#). In the Coax mode, use only the non-inverted SIO pin. AC-couple and terminate the inverting SIO pin using the series connection of a 100nF capacitor and a 49.9Ω resistor. In the STP configuration, both the non-inverted and inverted SIO pins are enabled by default.

Maximum cable length is limited by the frequency-dependent attenuation of the cable. Connectors degrade the return loss characteristic of the cable assembly. The GMSL2 channel specification allows two inline connectors, and provides detailed requirements for cable attenuation and return loss, as well as insertion loss and return loss requirements for PCB traces. In general, any physical channel implementation compliant with the GMSL2 channel specification can be used with reliable results. Contact the factory for the GMSL2 channel specification document.

### Selectable Line Rate

On power-up, the GMSL2 forward link rate is set by the CFG1 pin. See [Table 12](#). The initial configuration setting can be overridden by register writes.

The input ports, if applicable, can be independently configured to a different line rate using register writes.

### Adaptive Equalization (AEQ)

The GMSL2 devices automatically adapt the receiver's characteristics to compensate for the insertion and return loss characteristics of the channel that consist of the cables, connectors, and PCBs. This approach optimizes performance on any channel that meets the GMSL2 channel specification. The equalizer architecture makes GMSL2 links robust against noise, crosstalk, and reflections. Initial adaptation is performed during link lock and is then invoked at a rate of approximately 1Hz to track temperature and voltage variations. The adaptation process optimizes the equalizer coefficients to minimize the intersymbol interference observed at the output of the equalizer.

## Echo Cancellation

The GMSL2 link includes an echo cancellation circuit in both the serializer and deserializer to enable simultaneous transmission of high-speed video data and bidirectional control data.

## Spread Spectrum

Spread spectrum is used to reduce electromagnetic interference (EMI). Optional spread-spectrum clocking (SSC) is available to mitigate electromagnetic interference emitted from the device and interconnections and provides additional margin. SSC reduces peaks in the frequency spectrum by spreading the energy over a wider bandwidth. The forward-channel, spread-spectrum frequency is programmable in the range of 10kHz to 40kHz, and the frequency deviation is programmable in the range of 0% to  $\pm 0.125\%$ . If no forward-channel, spread spectrum is programmed, up to 0.5% frequency deviation can be tolerated.

## Line Fault

GMSL serializers include a novel line-fault detection circuit. It detects and reports open-circuit, short-to-battery, short-to-ground, and line-to-line short. The line fault monitor requires external resistors  $R_{EXT1}$  and  $R_{PD}$  in the Coax mode, and  $R_{EXT1}$ ,  $R_{EXT2}$ , and  $R_{PD}$  in the STP mode connected to the LMN<sub>x</sub> pins, as detailed in the following sections.

The line-fault monitor is disabled by default, and configuration options are available through registers, and status can be read by the register. If unmasked, a line-fault condition asserts ERRB. Line fault detection cannot be used in conjunction with Power-over-Coax (POC) or AC-coupled ground applications.

The line-fault monitor pins offer flexible connection and programming in either Coax or STP applications.

[Figure 19](#) illustrates the two configuration options to locate line-fault detection. The Local-Side Serializer Configuration is typically used for display links and Local-Side Deserializer Configuration for camera links. However, either configuration can be used on any device serial link system. Additionally, either applies to the Coax or STP modes.

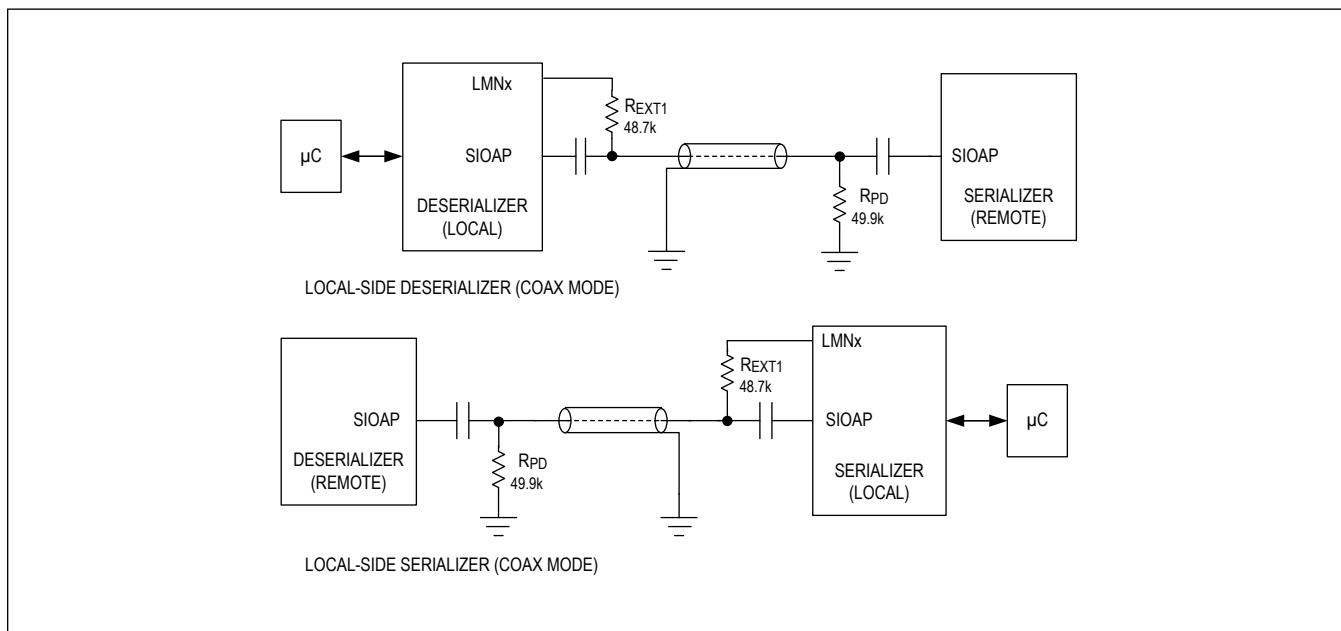


Figure 19. Line Fault Detection Location Options

The LMNx pins (LMN0 to 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.

## Coax Mode

In Coax applications, any LMNx pin may be used. The local side sources the line-fault. The local-side device requires a

single  $48.7\text{k}\Omega$  resistor ( $R_{EXT1}$ ) connected directly from any of the  $LMNx$  pins to the serial link.  $R_{EXT2}$  is not used in Coax applications. The remote side of the serial link requires a  $49.9\text{k}\Omega$  resistor ( $R_{PD}$ ) connected from the serial link to GND. Any of the line-monitor pins may be used in Coax applications.

**Table 5. Coax Mode Line Fault Configuration Options**

SIGNAL	SIOAP	SIOBP
Line Fault Pin	$LMNx$ (Any $LMN$ pin)	$LMNx$ (Any $LMN$ pin)
$R_{EXT1}$	$48.7\text{k}\Omega$	$48.7\text{k}\Omega$

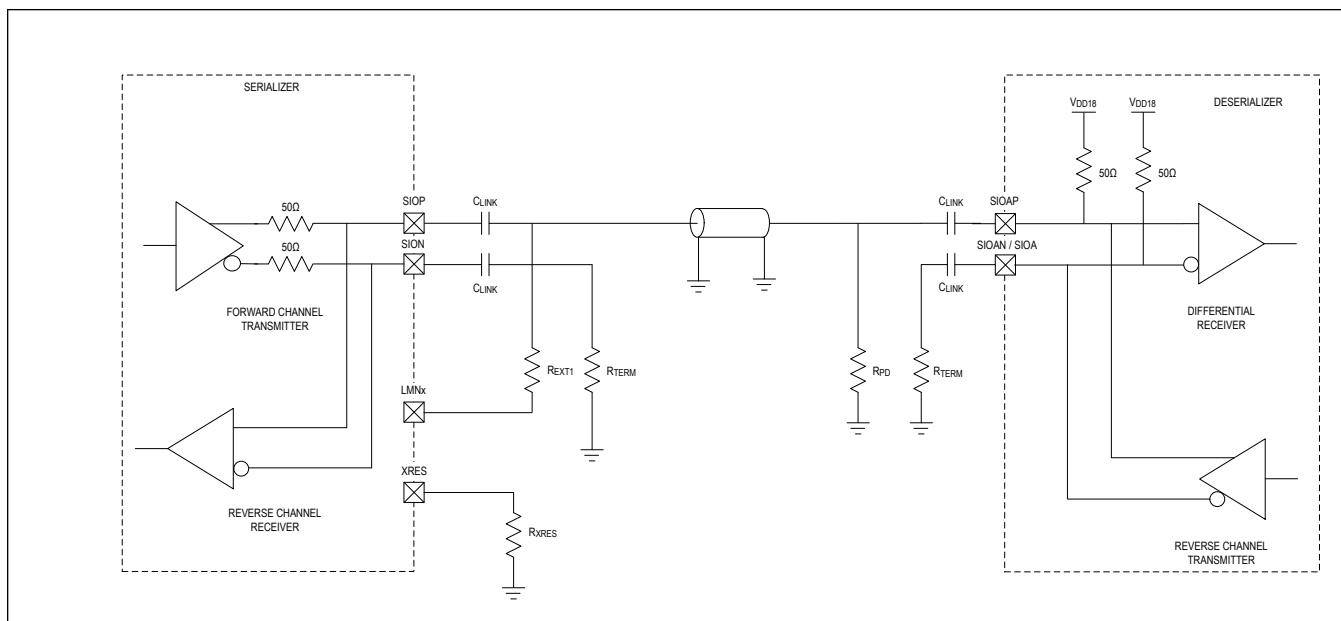


Figure 20. Typical GMSL Link Application Circuit for Coax Cable

### STP Mode

If the serial link is operating in the twisted-pair mode, connect one line to an even-numbered pin ( $LMN0/2$ ) and the other line to an odd-numbered pin ( $LMN1/3$ ) 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  $42.2\text{k}\Omega$  resistor ( $R_{EXT2}$ ); the odd-numbered pins (i.e.,  $LMN1$  and  $LMN3$ ) must use a  $48.7\text{k}\Omega$  resistor ( $R_{EXT1}$ ) to connect to the line (as with the single-ended mode). On the remote side, both lines require a  $49.9\text{k}\Omega$  resistor ( $R_{PD}$ ) connected to GND.

For full operation of line-fault detection in twisted-pair applications, ensure that  $LMN0/1$  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.

Line Fault Pair #1 ( $LMN0/LMN1$ ) Resistor Values and Connection Choices in STP Mode:

**Table 6. STP Mode Line Fault Configuration Options for  $LMN0/LMN1$**

LINE FAULT PAIR #1: $LMN0$ AND $LMN1$	SIOAP	SIOAN
Option #1	$LMN0: R_{EXT2} = 42.2\text{k}\Omega$ to serial link	$LMN1: R_{EXT1} = 48.7\text{k}\Omega$ to serial link
Option #2	$LMN1: R_{EXT1} = 48.7\text{k}\Omega$ to serial link	$LMN0: R_{EXT2} = 42.2\text{k}\Omega$ to serial link

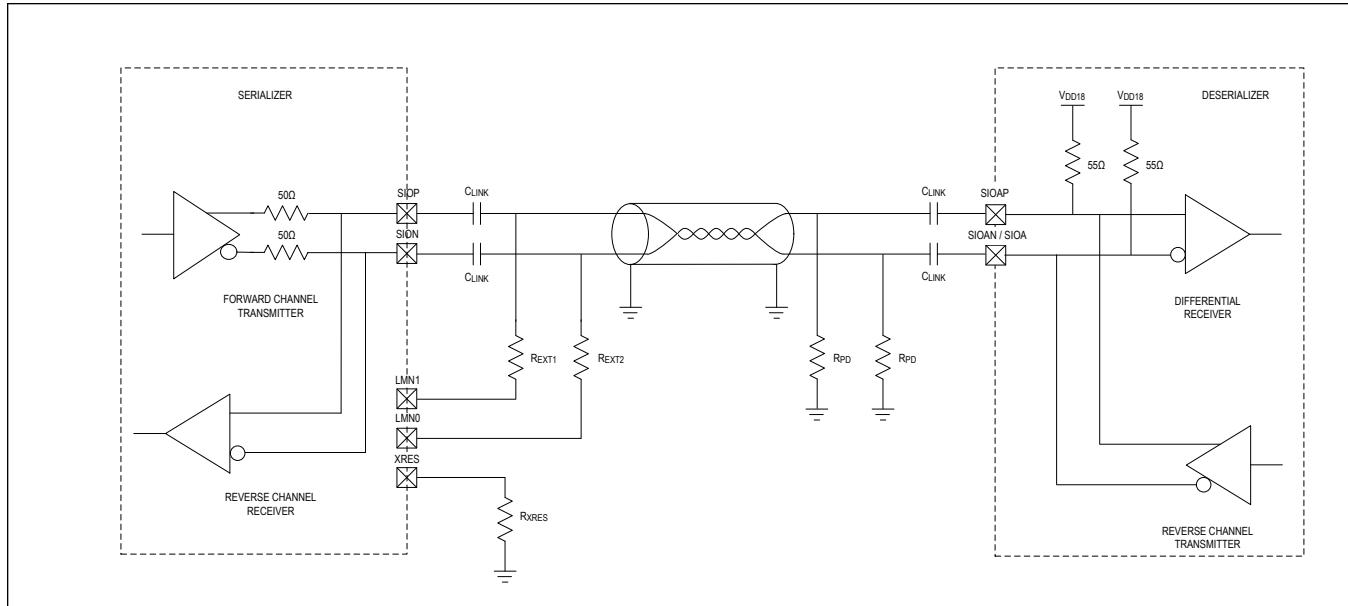


Figure 21. Typical GMSL Link Application Circuit for Twisted Pair (Line Fault Pair #1: LMN0 and LMN1, Option #2)

## Link Lock

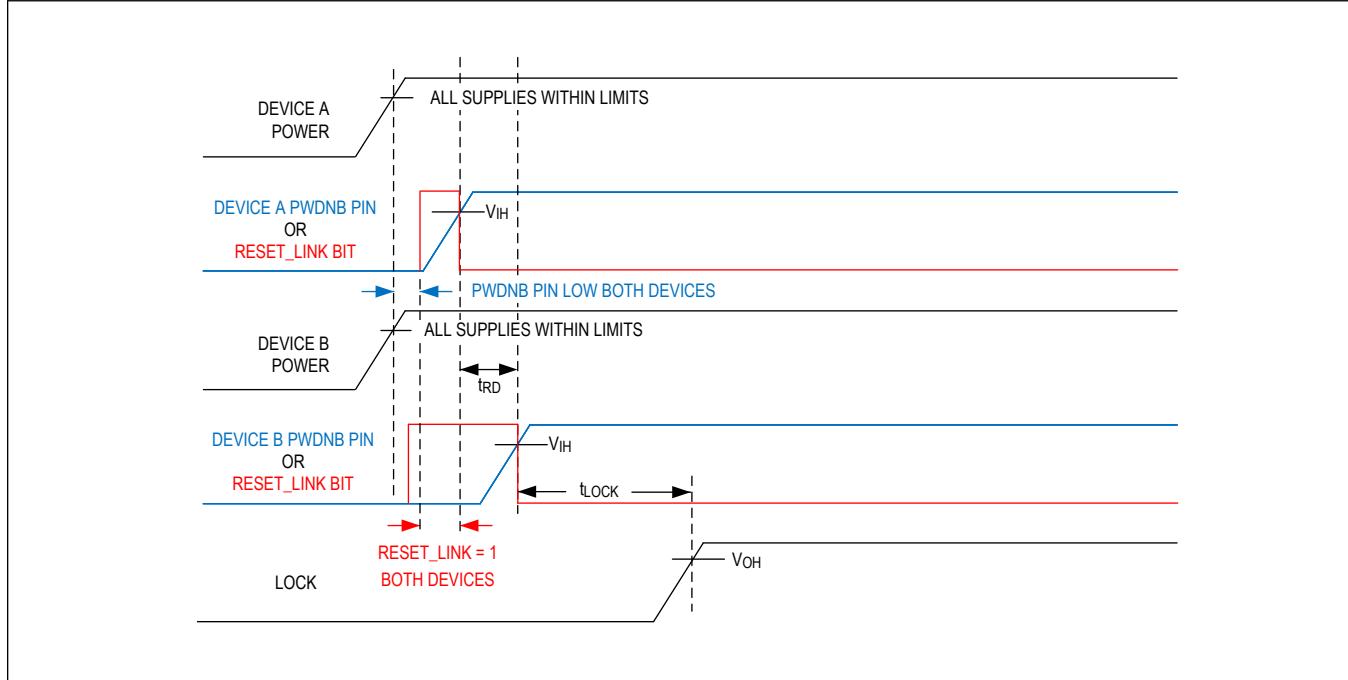


Figure 22. GMSL2 Lock Time

Figure 22 illustrates the sequence that is used to characterize GMSL2 link lock time. Device A is the first device (serializer or deserializer) to power up or resume operation from a RESET\_LINK state. Device B is the device (deserializer or serializer) at the other end of the GMSL link.

Link lock indicates that the data receive paths are locked (forward channel in the deserializer, reverse channel in the serializer). Video and control channel functions (I<sup>2</sup>C, GPIO) can be used immediately after link lock is asserted.

The device establishes single-link GMSL2 connectivity and link lock automatically following power-up. This is an indication that the cable is plugged in, and the system is up and running. Lock is obtained with no interaction between the µC and GMSL devices. Both serializers and deserializers have an open-drain LOCK output pin and a related status register.

The MAX96717 has only one link; therefore, it does not support dual-link and splitter configurations.

The GMSL2 link uses the crystal as the reference clock for GMSL2 links, so a valid video input (PCLK) is not needed for the GMSL2 link to lock.

#### Notes:

1. The lock sequence is initiated by the release of the PWDNB pin or the RESET\_LINK bit in either the serializer or the deserializer.
2. The lock time is measured from the PWDNB or the RESET\_LINK release, whichever is later, on either the serializer or the deserializer to the lock being asserted.
3. The PWDNB/RESET\_LINK states on the two sides of the link must have overlap when both the devices are in PWDNB/RESET\_LINK mode prior to the lock process starting.
4. If RESET\_LINK is used to initiate the lock, PWDNB is assumed to be high after power-up (normal operation).
5. If PWDNB is used to initiate the lock, RESET\_LINK is assumed to be low after power-up (normal operation).
6. Device A is the first device (serializer or deserializer) to be powered up. Device B is the device (deserializer or serializer) at the other end of the GMSL link.
7. To achieve the specified lock time, time delay  $t_{RD}$  (delay between the release of the PWDNB/RESET\_LINK on the two devices) must be less than 90ms. If this timing cannot be guaranteed, contact the factory for guidance.
8. Lock time and maximum allowed  $t_{RD}$  vary between different families of GMSL devices. They depend on the characteristics of both the serializer and deserializer. The typical lock time of a specific link can be best estimated as the longer of the lock times specified in each device data sheet. Similarly, the maximum permitted  $t_{RD}$  for a specific link can be estimated as the smaller of the values specified in each device data sheet. For further guidance, contact the factory.
9. If there is an instantaneous interruption to link lock, a period of 100ms following loss of lock should be provided to enable the link to automatically recover prior to any ECU initiated resets being issued. This will minimize any disruptions caused by a transient loss in connectivity.

#### GMSL2 Bandwidth Calculations

The GMSL2 forward link has a fixed link rate of 3Gbps or 6Gbps for the MAX96717. The reverse-link rate is fixed at 187.5Mbps. The GMSL2 protocol and channel coding overhead is roughly 16%. This leaves approximately 2.6Gbps or 5.2Gbps of video payload data throughput in the forward direction and 162Mbps in the reverse direction.

Ensure the worst cases do not exceed the available throughput of the forward and reverse links. Analog Devices' evaluation kit (EV kit) GUI includes a bandwidth (BW) calculator for initial bandwidth requirement estimates. Analog Devices also has other tools to calculate link-bandwidth utilization. Consult the factory for high-bandwidth use cases to ensure error-free performance.

[Table 7](#) provides rough estimates of the bandwidth utilization for each communication channel.

**Table 7. Forward- and Reverse-Link Bandwidth Utilization**

DATA	APPROXIMATE BANDWIDTH UTILIZATION
Video (Forward Path Only)	<p>Bandwidth = <math>PCLK \times (bpp + 1 + \text{video\_pixel\_CRC}) \times 10/9 \times 2048/2047 \times 128/120^*</math></p> <p>Pixel mode PCLK Calculations:</p> <p><math>PCLK = \text{MIPI data rate}/\text{bpp}</math></p> <p><math>PCLK = \text{MIPI data rate}/(2 \times \text{bpp})</math> for double pixel mode</p> <p><math>PCLK = \text{MIPI data rate}/(3 \times \text{bpp})</math> for triple pixel mode</p> <p>Tunnel mode PCLK calculations:</p> <p><math>PCLK = \text{MIPI data rate}/24</math></p> <p>Bandwidth Notes:</p> <ol style="list-style-type: none"> <li>1. The link bandwidth calculation uses the bpp value of the video pipe with highest bpp value of the transmitted datatype(s) (Pixel mode only).</li> <li>2. Maximum bandwidth is limited by pixel clock rate PCLK.</li> <li>3. <math>\text{video\_pixel\_CRC}=0.5</math> (when video pixel CRC is enabled)</li> </ol> <p>PCLK Notes:</p> <ol style="list-style-type: none"> <li>1. The PCLK calculation uses datatype with the lowest bpp value (Pixel mode only).</li> <li>2. Maximum PCLK is 300MHz for 3Gbps link rate</li> <li>3. Maximum PCLK is 600MHz for 6Gbps link rate</li> <li>4. MIPI data rate includes horizontal and vertical blanking times (*only applies when GMSL FEC is enabled)</li> </ol>
I <sup>2</sup> C	13 to 40 x I <sup>2</sup> C clock rate
UART	6 x UART bit rate
SPI	1.7 to 3.1 x SPI rate, depending on SPI byte length
GPIO	60 x GPIO transition rate without delay compensation 80 x GPIO transition rate with delay compensation enabled

**Note:** Bandwidth utilization for all video and control-channel communication in the forward direction increases by 6.7% when forward-error correction (FEC) is enabled.

Definitions:

H = Horizontal resolution (active pixels)

V = Vertical resolution (active video lines)

fps = frames per second

bpp = bits per pixel

MIPI data rate = Aggregate data rate of all lanes in the mobile industry processor interface (MIPI).

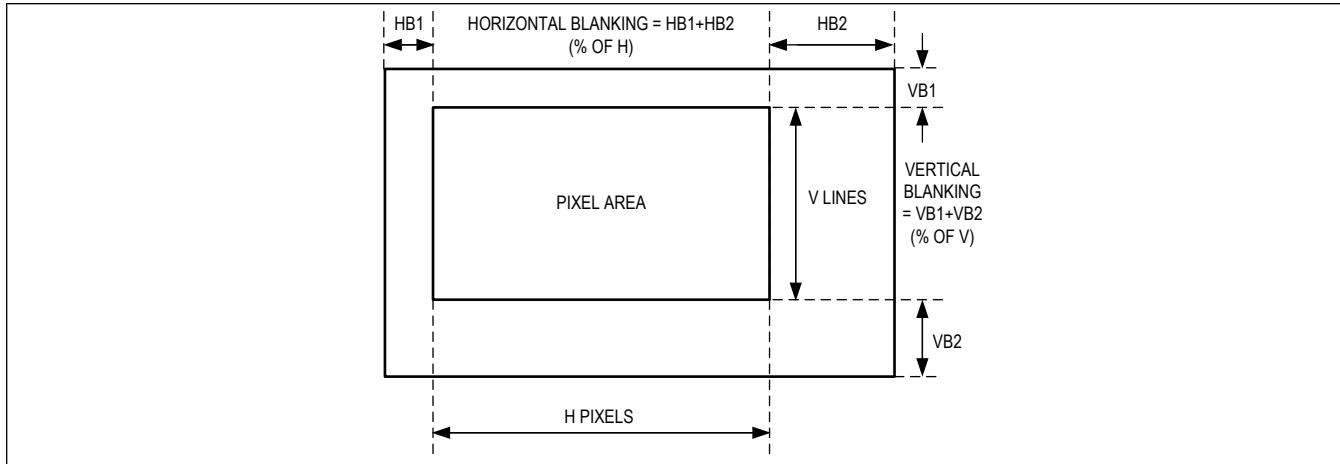


Figure 23. Video Frame Format for Bandwidth Calculation

## Control Channel and Side Channels

A  $\mu$ C or other controller can send and receive control and side-channel data over the GMSL2 serial link simultaneously with high-speed video data. Most GMSL2 devices support the following interfaces:

- Main I<sup>2</sup>C/UART (internal access)
- Pass-through I<sup>2</sup>C/UART
- SPI
- GPIO

All of the above interfaces can pass data through the GMSL2 link, but the GMSL2 device registers can be accessed and configured only through the main I<sup>2</sup>C/UART interface, unless the pass-through I<sup>2</sup>C/UART ports are enabled through register control.

The side channel, with its various interfaces, is accessed using multifunction pins. Multifunction pins have a default function and can be programmed to an alternate function after power-up. Due to a practical limit in the number of pins available on a given device, not all interfaces can be simultaneously supported. See [Pin Descriptions](#) and [Table 13](#) for default and alternate multifunction pin functions, as well as available combinations of interfaces.

### Main I<sup>2</sup>C/UART

The main I<sup>2</sup>C/UART is located on the SDA\_RX and SCL\_TX pins of each GMSL2 device. The I<sup>2</sup>C (SDA, SCL) or UART (Tx, Rx) interface is selected by the CFG0 pin voltage at power-up (see [Table 11](#)). The selected interface provides main access to both GMSL2 registers and device registers from either end of the link.

The main microcontroller ( $\mu$ C) can reside on either end of the link (usually the serializer side for display applications and deserializer side for camera applications). The MAX96717 supports dual-main microcontrollers, provided that software arbitration, such as token passing, is used to prevent packet collisions. In addition, while utilizing dual-main configuration, DIS\_Rem\_CC can be used to disable remote control channel configurations to avoid bus contention. The control channel allows only one main  $\mu$ C to communicate at a time.

To configure peripheral devices over the link, the GMSL2 serializer and deserializer must use the same control-channel interface (both I<sup>2</sup>C or both UART). Unlike legacy GMSL1 devices, there is no I<sup>2</sup>C-to-UART conversion capability. I<sup>2</sup>C /UART outputs are open-drain and require appropriately-sized external pullup resistors for proper operation.

For detailed main channel programming information, see the [Control-Channel Programming](#) section under [Applications Information](#).

### I<sup>2</sup>C/UART CRC and Message Counter

The MAX96717 provides additional functional safety by adding an optional CRC to I<sup>2</sup>C/UART read/write transactions, and a separate message counter for read and write packets. These features are either enabled by default at power-up or enabled by register settings. The CRC and message counter can be used together or individually. The CRC and

message counter features only apply to device register read/write transactions, and are not supported for pass-through traffic. The CRC and/or message counter can be enabled in the serializer, deserializer, or both.

The first CRC byte covers the data for the first 6 bytes: Device Address, Register Address MSB, Register Address LSB, Message Counter MSB, Message Counter LSB, and the first data byte. If there are multiple data bytes after the first byte, then a CRC byte is appended after each subsequent data byte. The CRC engine is reset to 0 after each CRC calculation is complete, meaning a new CRC calculation is done for each additional byte. The CRC polynomial for I<sup>2</sup>C/UART is  $P(x) = x^8 + x^6 + x^3 + x^2 + 1$ .

### I<sup>2</sup>C/UART CRC and Message Counter Options

At power-up, the I<sup>2</sup>C/UART CRC and Message Counter features are disabled by default. Both features can be enabled or disabled through register control.

### Pass-Through I<sup>2</sup>C/UART

The MAX96717 has two pass-through I<sup>2</sup>C/UART channels. These channels do not have access to registers in either the GMSL2 serializer or deserializer; they simply tunnel the I<sup>2</sup>C or UART signal across the GMSL2 link. This allows I<sup>2</sup>C channels to be separated so that no multimain conflicts occur. I<sup>2</sup>C/UART outputs are open-drain and require appropriately-sized external pullup resistors for proper operation.

### Serial Peripheral Interface (SPI)

The MAX96717 enables a host SPI main on one side of the GMSL2 link to control a peripheral SPI subordinate on the opposite side. Communication may be in either direction across the GMSL2 link. Although multiple SPI peripherals may be connected, it is recommended that only one be communicated with at a time. Contact the factory for guidance on configuring SPI connections

The SPI clock range is 600kHz to 25MHz. Care must be taken to meet setup and hold-time requirements when using at speeds higher than 20MHz. The speed rating for each MFP needs to be set correctly for reliable operation. [Table 8](#) specifies the recommended drive strength and latching edge for the SPI as a function of serial clock (SCLK) speed.

**Table 8. SPI Latching Edge and Speed**

FREQUENCY	V <sub>DDIO</sub>	LATCHING EDGE	PIOX_SLEW[1:0]
<12.5MHz	1.7V to 2.24V	Opposite from Shift	01
	2.25V to 3.6V		10
12.5MHz to 25MHz	1.7V to 2.24V	Opposite from Shift	00
	2.25V to 3.6V		01

### General-Purpose Input and Output

The MAX96717 provides up to 11 GPIO/GPO/GPI, dependent on device feature utilization. GPIOs are typically used to tunnel low-speed (< 100Kbps) signals over the GMSL2 link. A GPIO tunnel can be set up in the forward or reverse direction. MFP pins can be programmed as GPI, GPO (push-pull output), or ODO (open-drain output).

Each GPIO pin can be configured as an input, output, or input/output by programming the GPIO\_TX\_EN and GPIO\_RX\_EN register bits of each GPIO pin. Note that unwanted loop behavior is avoided for pins configured as input/output; when the pin is driven by a transition received from the remote side, the driven GPIO transition is not transmitted back. GPIO pins may alternately be controlled and read solely from registers.

Most GPIO pins can be programmed for 1MΩ or 40kΩ pullup or pulldown (or none).

When a GPIO is programmed as GPO, the GPO can generally be programmed for open-drain or push-pull output.

A GPIO packet has 32 possible GPIO channel IDs. Each GPI is mapped to a channel ID according to the GPIO\_TX\_ID register. On the receiving end, each GPO outputs the received data with a programmed GPIO channel ID corresponding to the GPIO\_RX\_ID register for that pin. This provides flexibility in determining which GPIO input drives which GPIO output.

GPIO transmissions are transition-based; a GPIO packet is created and transmitted on the GMSL2 link when a rising or falling edge transition is detected at a GPIO pin. Several GPIO transitions at different GPIO pins can be grouped into a single packet. These pin transitions can be transmitted in two different modes: regular and delay-compensated.

The GPIO channel is not bandwidth-efficient and should be used for low-speed signals only. Each GPIO transition uses 40 bits to 80 bits on the GMSL2 link for transmit and 40 bits to 60 bits on the reverse (due to received ACK packets). Bandwidth usage values vary based on channel configuration: ARQ enable, CRC enable, and double-header enable, all impact channel bandwidth usage.

The state of each GPIO can be read or written by register, either locally or remotely, over the GMSL2 link by a  $\mu$ C using the control-channel I<sup>2</sup>C/UART interface. In non-delay-compensated mode, channel latency is not fixed. The GPIO transition is sent as soon as possible, based on priority and available link bandwidth. This variable delay is a result of multiple communication channels sharing the link. Non-compensated mode should be used with signals tolerant to delay variation (i.e.,  $\mu$ C interrupts). Priority can be set for GPI pins using registers. If no priority is set, GPI transitions are transmitted in the order they occur. However, when priority is set, transitions on GPI with higher priority are transmitted earlier.

Typical GMSL2 device delays for forward-link and reverse-link rates are shown in [Table 9](#).

**Table 9. Typical GPIO Delays for Forward-Link and Reverse-Link Transmission**

DIRECTION	DELAY COMPENSATION	DELAY
GPIO forwarding from serializer to deserializer	0	1 $\mu$ s
	1	3.5 $\mu$ s
GPIO forwarding from deserializer to serializer	0	6 $\mu$ s
	1	15 $\mu$ s

### Delay in Non-Compensated Mode

In Non-compensated mode, the value of the transition is transmitted along with the GPIO channel ID. Note that GPIO channel latency is not fixed; the GMSL2 link has variable delay as a result of multiple communication channels sharing the link. A maximum latency limit is established by the GMSL2 bandwidth-sharing scheme, but significant fluctuation remains. Non-compensated GPIO mode should be used with signals invariant to the delay variations (e.g.,  $\mu$ C interrupts).

### Delay-Compensated Mode

In Delay-compensated mode, a timestamp value is transmitted in addition to the value of the transition and GPIO channel ID. This timestamp is a high-resolution value sampled by an internal 600MHz clock that records when the GPIO transition is detected at the input. The remote-side chip uses the timestamp value to wait and output the GPIO transition after a total fixed delay from the GPIO input transition. This method mitigates possible variable latency issues by making the total GPIO input-to-output delay a precise, fixed value. Delay-compensated GPIO mode should be used for signals of which the relative timing of rising and falling edges are important (e.g., pulse-width modulation (PWM), camera frame sync, radar ramp trigger, and low-speed UART signals).

### Frame Sync

In surround-view camera applications, a frame-sync signal is usually required by the sensors to synchronize the output of a frame with the other cameras in the system. Most GPIOs can be configured as GPI and linked to a frame-sync signal generated by a surround-view camera electronic control unit (ECU).

### Functional Safety Features

The MAX96717 integrates a number of safety features, including power-on-self-test (POST) that includes logic built-in self test (LBIST) and memory built-in self test (MBIST), and a power manager that reports undervoltage/overvoltage conditions. The host interface (I<sup>2</sup>C/UART) has an optional CRC capability and optional message counter for I<sup>2</sup>C/UART traffic.

At power-up, LBIST verifies that key logic blocks are correctly functioning and free of latent faults. MBIST checks the memories for defects on power-up. The POST produces a pass/fail result that can be read through a register for each test. A POST failure on start-up does not prevent the device from operation but is reflected in registers POST\_LBIST\_PASSED and POST\_MBIST\_PASSED.

A register CRC (REGCRC) detects unintended changes in the configuration registers. A CRC value for the configuration registers is computed and stored on demand at start-up. During operation, the REGCRC block periodically calculates the

CRC of the configuration registers and compares the calculated value to the stored value. A mismatch can be reported through ERRB. Refer to the product specific safety manual for a complete list of safety features.

### Analog-to-Digital Converter

The MAX96717 features a 10-bit, integrating analog-to-digital converter (ADC) with a multiplexed, single-ended input to monitor DC-voltages. The multiplexed input allows the monitoring of external input lines (ADC0, ADC1, and ADC2), and internal power supplies. The input range of the ADC extends from ground to a full-scale voltage, dependent upon the selected reference voltage. The default is a built-in 1.25V voltage reference (preferred). Alternatively, an external 1.25V precision reference or the  $V_{DD18}$  supply voltage divided by 2 ( $V_{DD18}/2$ ) can be used. With the 1.25V reference, the maximum ADC input voltage is 1.2V. When using  $V_{DD18}/2$  as the reference, the maximum input voltage is limited to  $V_{DD18}/2$ . Higher input voltages are supported using built-in programmable voltage dividers at the ADC input pins.

The ADC is controlled through the MAX96717 register map with feedback through interrupt flags. Controls include ADC power-up control, input channel selection, overrange/underrange thresholds, and conversion start. Feedback includes ADC conversion done, ADC ready, ADC overflow, and conversion overrange and underrange. Input and voltage monitoring, and threshold levels are set up through the register map.

The overvoltage/undervoltage thresholds can be programmed for up to eight individual ADC input channels. These thresholds can be enabled to assert an error flag if the converted ADC value is out of range. This feature allows voltage monitoring without the need to read out the converted value after each A/D conversion. This allows to issue a conversion, review the error flag status for overrange/underrange, and proceed to the next channel, when ready.

The ADC also contains a round-robin state machine that continuously cycles through up to eight input channels for continuous voltage monitoring. When an overrange/underrange or overflow occurs, an error flag asserts.

The state machine also contains an ADC shutdown mode that powers off the ADC for up to 65,535 ADC conversion cycles.

The ADC contains a low-offset input buffer that should always be used. The minimum input voltage for buffered input is 100mV, below which the ADC shows some nonlinearity affecting its accuracy. The conversion time is approximately 430 $\mu$ s. Do not read the ADC conversion done interrupt flag (adc\_done\_if) before the conversion is done.

### ADC Features

- Input buffer, ADC, and voltage reference.
- Programmable input multiplexer with three channels each to monitor external and internal supply voltages.
- Integrated voltage dividers (/1, /2, /3, and /4) at the MFP pins to measure voltages as high as  $V_{DDIO}$ .
- Input for optional external precision voltage reference.
- Overvoltage and undervoltage interrupt generation.
- ADC conversion start and done interrupt generation.
- Power control with interrupt generation when ADC and charge pump power-up are complete.
- Continuous round-robin monitoring of all channels.
- Programmable input and reference gains to scale the input and reference.

### ADC Architecture

The ADC on the MAX96717 is an integrating type analog-to-digital converter with a single-ended input multiplexer, high-impedance input buffer, and integrated reference generator. See [Figure 24](#). The input buffer is high-input impedance and it minimizes the loading of high-impedance external inputs.

The ADC reference voltage can be based on the built-in 1.25V reference (preferred), the  $V_{DD18}$  supply level divided by 2, or an external pin ( $V_{REF}$ ).

Optimum accuracy is achieved by connecting an external reference to the  $V_{REF}$  pin while powering down the internal-reference buffer.

The ADC supports an input gain of 1x (unity) or 2x. For signals less than half the voltage reference, the 2x gain can provide additional resolution.

The ADC input features a 16-channel input multiplexer. However, only six physical channels are used (0, 1, 2, 8, 9, and A). These physical channels can be mapped to any of the ADC logical channels.

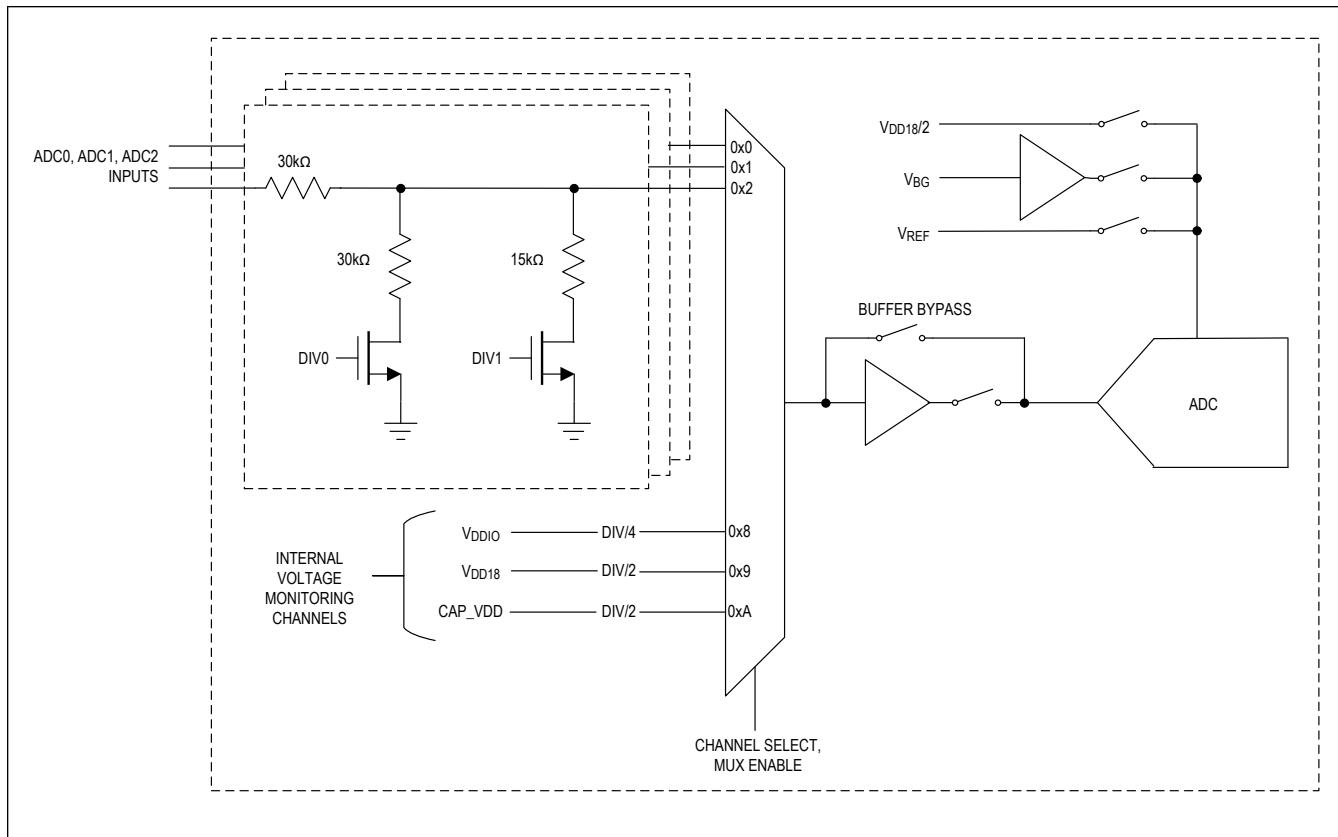


Figure 24. ADC Block Diagram

## ADC Operation

### ADC Control Interface

The control registers for the ADC are used to set up the ADC, start A/D conversions, and retrieve the results of ADC conversion operations. The ADC control interface enables:

- Configuration of the input multiplexer, reference, input buffer, and ADC.
- Power control and generation of interrupt when the charge pump power-up delay is complete.
- A/D conversion start and done interrupt generation.
- ADC threshold limits and interrupt generation.

After an A/D conversion is complete, the data is available to be read from the ADC\_DATA register. The ADC output value is a function of the input voltage, reference voltage, and optional data scaling. The voltage reference for the conversion can be the internal 1.25V reference (preferred), an external reference, or VDD18 voltage supply. The ADC input buffer should be enabled. Bypassing the input buffer is not recommended.

Interrupt flags are used to provide status and error conditions. Individual interrupt flags are set to indicate ADC Ready (power-up complete), ADC Done (conversion done), ADC Low-Range violation, ADC High-Range violation, and ADC Overflow. Associated with each channel is a channel low- and high-range violation interrupt flag.

Each interrupt flag has an enable bit to indicate it should be monitored. When a flag is set by the ADC, it remains set until it is cleared, by reading the interrupt register associated with the asserted flag.

The ADC can also operate in a round-robin fashion to continuously monitor up to eight channels of the input MUX. Each channel contains individual controls for monitoring. Upper and lower limits can be set individually for each channel, and each channel can be individually configured as an interrupt source.

During the round-robin, the ADC Done and ADC Ready interrupt flags are masked.

### ADC Power-Up and Power-Down Sequences

The MAX96717 must have  $V_{DD}$ ,  $V_{DD18}$ , and  $V_{DDIO}$  supplies available, and the link between the serializer and deserializer must be established before the ADC can be used.

After the power supplies are validated, the ADC clock must be enabled and power-up controls applied. Note that the ADC's charge pump requires a 10 $\mu$ s delay from initial power-up before it is in steady-state and available for use. The interface detects when the ADC circuits are enabled and sets the ADC ready interrupt flag. Once the delay is passed, the ADC is ready for use. See ADC\_CTRL and ADC\_INTR register documentation for further details. Refer to the GMSL2 User Guide for details of the sequence of steps to power-up, use, and power-down the ADC.

### ADC Overvoltage and Undervoltage Thresholds

To simplify power supply monitoring, the ADC supports programmable data limits and interrupt enables for all channels. At the end of conversion, the ADC output is compared to a limit when a selected channel is sampled. If the detector and overlimit interrupt are enabled, an interrupt is generated. This same capability is provided for the underrange limit detector. Refer to the GMSL2 User Guide for details on programming.

### ADC Round-Robin Sampling

The MAX96717 can continuously monitor the enabled channels by cycling through them. The ADC can also be programmed to automatically enter Sleep mode to shut down for up to 65,535 ADC conversion cycles.

### ADC Application Diagrams for External Voltage Measurements

Using the ADC to measure external voltages requires planning to understand how the input network of the ADC0, ADC1, and ADC2 pins are implemented.

For voltages less than  $V_{DDIO}$  (maximum voltage of the ADC input), the internal dividers can be used to bring the voltage to within the valid input range of the ADC. For voltages greater than  $V_{DDIO}$ , an external resistor divider can be used to create a voltage within the input range of the ADC. When external dividers are used, the ADC input resistor-divider should be programmed for /1 mode and the internal buffer enabled.

External resistors in series with the internal dividers should not be used because large variations in voltages are experienced due to on-chip resistor variations.

### Measuring External Voltages Lower than $V_{DDIO}$

When measuring voltages below 3.6V, as shown in [Figure 25](#), the internal dividers can be used to adjust the voltage into the ADC to be  $\sim 1V$  (or  $V_{REF}$ ). [Table 10](#) shows the recommended divider setting for  $V_{IN}$  ranges (Note: In [Table 10](#), <ch> refers to the ADC logical channel). Note that the maximum input voltage to the ADC should not exceed  $V_{DDIO}$ . If a higher voltage needs to be monitored, it should be divided down using an external resistor-divider.

**Table 10. ADC Voltage Divider Settings**

$V_{IN}$ VOLTAGE RANGE	DIVIDER PROGRAM	ADC_CTRL_3.adc_div or ADC_LIMIT<ch>_3. div_sel<ch>
$< V_{REF}$	/1	2'b00
$V_{REF}$ to 1.8V	/2	2'b01
1.8V to 3V	/3	2'b10
3V to 3.6V	/4	2'b11

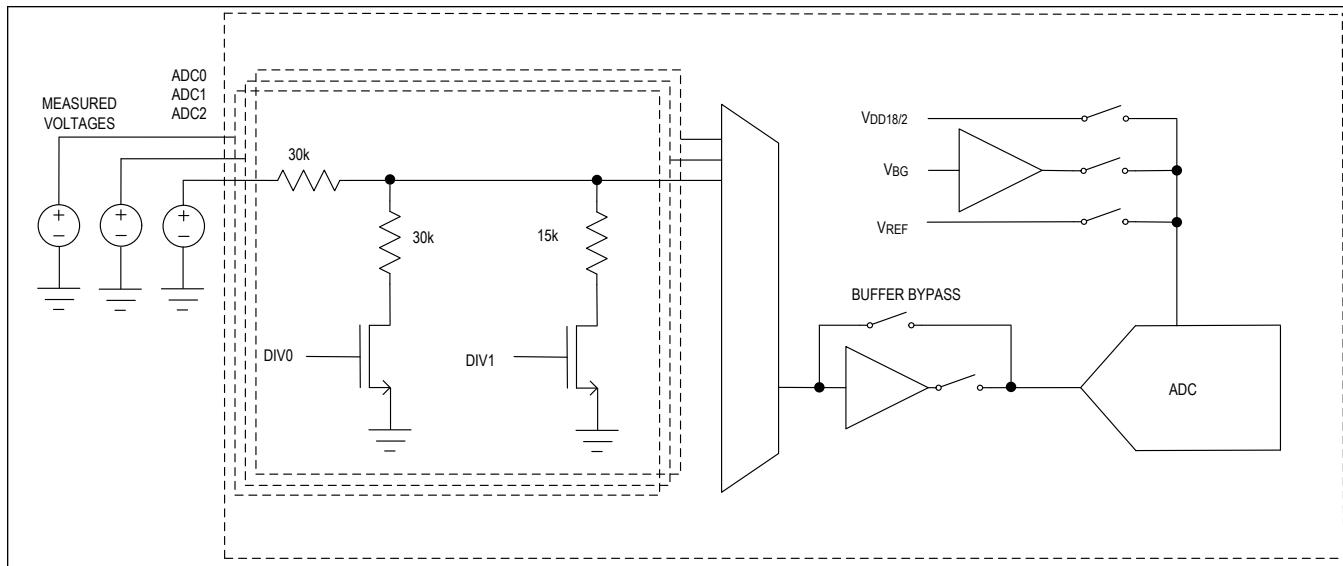


Figure 25. Measuring External ADC Voltages < 3.6V

### Measuring Voltages Higher than $V_{DDIO}$

To make measurements of these voltages higher than 3.6V, an external resistor-divider should be used to bring the voltage within the range of the ADC ( $< V_{REF}$ ). See [Figure 26](#). In this application, program the internal divider setting to /1 mode (2'b00).

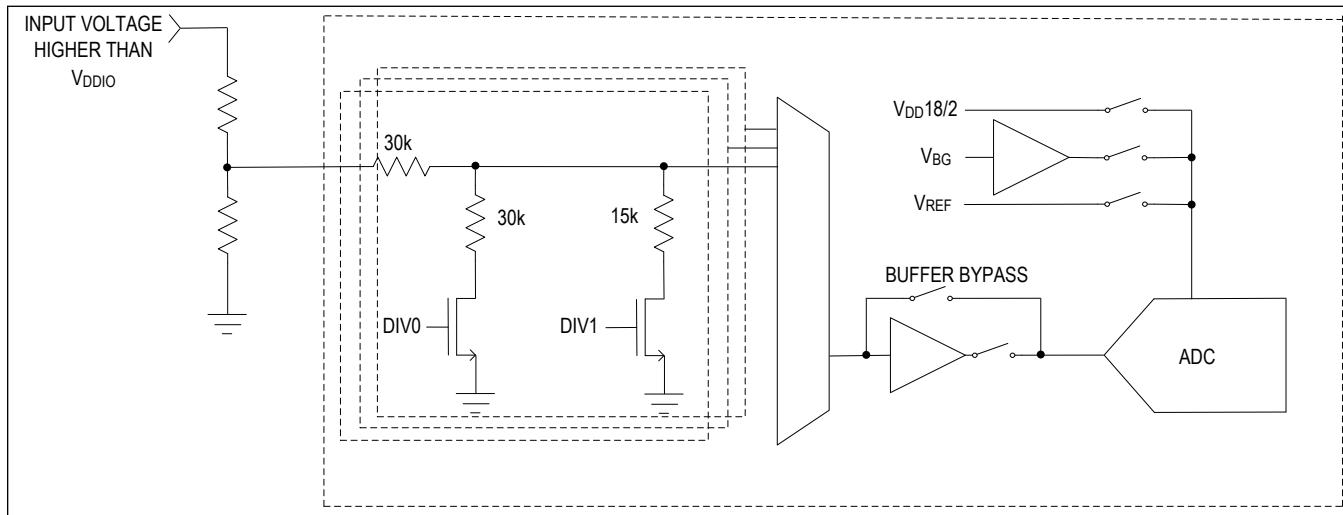


Figure 26. Measuring External ADC Voltages Greater Than  $V_{DDIO}$

### Junction Temperature Monitoring and Overtemperature Alarm

The MAX96717 includes redundant temperature sensors to measure the die temperature. The junction temperature of the MAX96717 can be monitored by initiating a junction temperature measurement and reading the junction temperature out through registers (REGADCBIST0/13/14/15). The junction temperature is reported in degrees Kelvin with 0.5° resolution. Overtemperature thresholds can be created and monitored by round-robin state machine or system-on-chip (SoC). Die temperature thresholds need to be set equal to +125°C (or lower, if preferred).

## Eye-Opening Monitor

The eye-opening monitor (EOM) enables GMSL2 parts to monitor the link margin on an active link and generate an interrupt if it falls below an acceptable level. For example, if a cable is damaged, the link can run error-free, but have less link margin than desired. This allows the customer to react proactively to deteriorating cable performance before any link errors occur. GMSL2 parts can measure the horizontal or vertical eye-opening of the equalizer's output. The measurement is activated automatically at a rate of approximately 1Hz once a link is active. The EOM block compares the data sampled at the center of the eye with a sample offset in phase (for the horizontal EOM) or offset in voltage (for the vertical EOM). An eye-opening figure of merit is then reported. The EOM can trigger an interrupt or a reset if the opening falls below user-defined thresholds.

## Sleep Mode

The sleep state preserves critical register settings and configurations. Retained registers are detailed in the register table. The device can be put into the sleep state through an I<sup>2</sup>C/UART command. Resume is invoked by an I<sup>2</sup>C command or a low-frequency clock beacon transmitted from the main device over the GMSL2 link. See the Register Map to determine if any register writes are needed to fully restore the device to the presleep condition. Note that GPIO levels set by received values from the other side of the link are not retained when entering sleep mode. Disable GPIO receive and set GPIO values to known levels before entering the sleep mode.

## Other Functions

GMSL2 serializers and deserializers have a main I<sup>2</sup>C/UART control-channel interface that a μC uses to access serializer and deserializer registers, as well as peripheral devices, from either end of the link. Each device also has two pass-through I<sup>2</sup>C/UART channels available for local or remote peripheral control. The pass-through I<sup>2</sup>C/UART channels do not have access to serializer and deserializer registers.

The MAX96717 also includes an SPI main/subordinate with two subordinate-select pins for peripheral control. The SPI enables a host SPI main on one side of the GMSL2 link to control a peripheral SPI subordinate on the opposite side. The host can be located at either end of the link or can swap ends by reprogramming the GMSL2 devices, as a GMSL2 device can be configured as an SPI main or subordinate.

The MAX96717 provides up to 11 GPIOs dependent on device feature utilization. GPIOs are typically used to tunnel low-speed (< 100Kbps) signals over the GMSL2 link. A GPIO tunnel can be set up in the forward or reverse direction.

The devices include a video crossbar. The crossbar can be used to reorder the color and sync signals. It can also be used for D-PHY lane remapping and phase inversion, if desired.

GMSL2 devices incorporate numerous link-margin optimization and monitoring functions to ensure a high link margin. Adaptive equalization periodically (~1Hz) optimizes the link margin to adapt to environmental changes and cable aging. An eye-opening monitor function for continuous link-margin diagnosis with various threshold alarm levels is available for runtime alerts of link degradation. A pseudorandom bit sequence (PRBS) checking function is available to verify the correct link and video-channel operation.

## Video PRBS

The video channel of the serializer includes a PRBS pattern generator that operates with bit-error verification in the deserializer to test channel operation. The PRBS generator works with the clock received from the source.

To run the video PRBS test, first enable the PRBS generator on the transmitter side, then enable the checker on the receiver side. The PRBS checker automatically synchronizes itself to the incoming PRBS pattern. If it is unable to synchronize within a few cycles, it asserts the PRBS\_FAIL register. To stop the PRBS test, first turn the checker off on the receiver side, then turn the generator off on the transmitter side.

PRBS errors can be read from the VPRBS\_ERR register on the deserializer side. Any PRBS error causes the ERRB pin to go low in the deserializer by default (see VPRBS\_ERR\_OEN and VPRBS\_ERR\_FLAG register bits).

**Note:** The video channel shares link bandwidth; it is possible to have a bit error on the link, which does not cause a video PRBS error.

## MIPI CSI-2 Input Interface

### Lane Configurations and Data Rates

Image data is received on a unidirectional MIPI CSI-2 v1.3 input port. The port should be configured to use D-PHY v1.2. The interface offers a high degree of flexibility. When configured as D-PHY, the interface supports either one, two, three, or four differential lanes. To ease the PCB layout, lane swapping is supported independently of how many lanes are used. For example, when the device is configured to enable a single lane, that lane can be any one of the four lanes available within that port. If two lanes are enabled, those two lanes can be any two lanes within the port. Lane polarity swapping within a lane is also supported. [Figure 27](#) shows the options and default mapping. D-PHY mode supports data rates up to 2.5Gbps per lane.

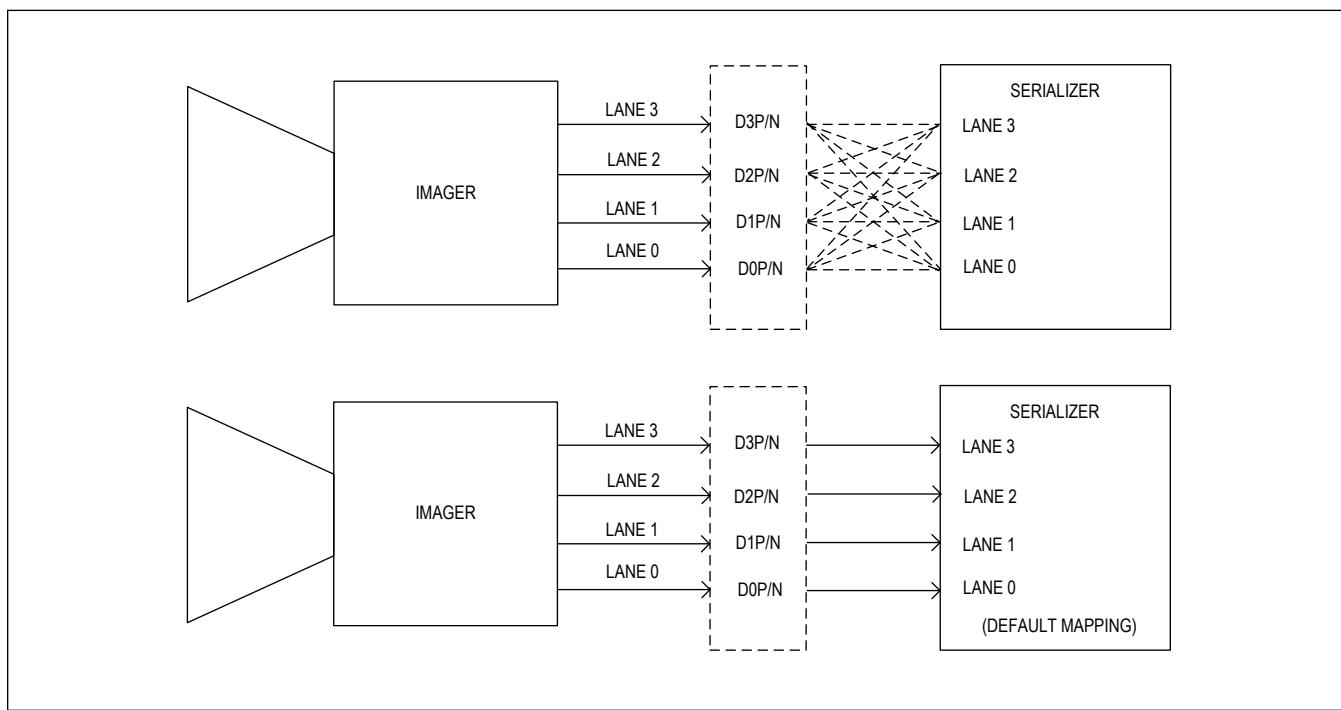


Figure 27. MIPI CSI-2 Lane Mapping Options and Default Settings

### Clocking

The device CSI-2 interface supports both continuous and burst-mode clocking in the D-PHY mode.

### CSI-2 Virtual Channel and Data Type Interleaving

The device supports virtual channels, including Virtual Channel Extension (VCX) introduced in CSI-2 v2.0, enabling up to 16 virtual channels.

If operating in the pixel mode, the virtual channel and/or data type received from the CSI-2 source can be reassigned (pixel mode only) or passed through (pixel and tunneling modes).

### Lane Deskew

The CSI-2 D-PHY interface can be configured to use interlane deskew using deskew patterns from the transmitter when the bit transmission rate is 1.5Gbps/lane and above. Deskew is optional for data rates lower than 1.5Gbps/lane. Deskew is initiated by the transmitter under CSI-PPI control.

### Minimum Blanking

The minimum horizontal blanking period needed by the CSI-2 serializers and deserializers is the maximum of either 40 pixels or  $300\text{ns} + 370\text{UI}$  (where UI is defined as the period of CSI-2 lane rate). For most cases, 40 pixels is the larger number. Minimum vertical sync width is one video line. Minimum vertical front porch is one video line. Recommended vertical back porch is one video line.

Minimum vertical back porch in pixel mode is the maximum of:

- 40 pixels
- $300\text{ns} + 370\text{UI}$

Minimum vertical back porch in tunneling mode is the maximum of:

- 40 pixels
- $200 \times \text{PCLK}$  periods + 233ns, where  $\text{PCLK} = \text{total MIPI data rate}/24$
- $300\text{ns} + 370\text{U}$

These calculations provide rough estimates. Use the GMSL GUI bandwidth calculator for more precise values.

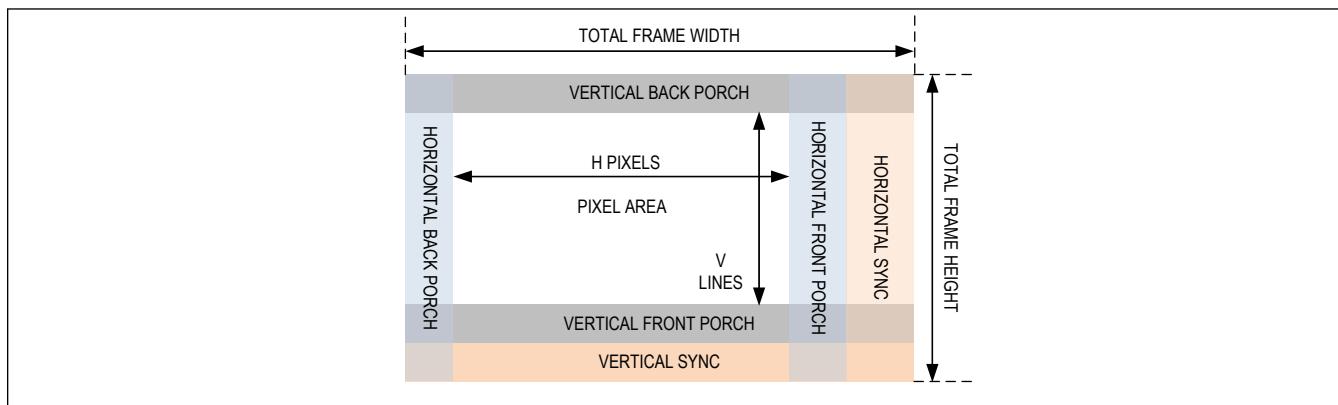


Figure 28. Video Timing

### MIPI End-to-End Packet Spacing

When in tunneling mode, timing requirements must be met to avoid a false line-CRC error flag. As shown in [Figure 29](#), the end-to-end packet spacing (L1, L2, ... LN, LE, LS) must be a minimum of  $200 \times \text{PCLK}$  cycles + 233ns, where PCLK is the total MIPI data rate/24. This limit is typically a concern for the line-end short packets that follow the last-long data packet of a frame. An alternate solution is to disable the line-CRC check and rely on the MIPI CSI-2 packet CRC, which is a valid verification of error-free data reception.

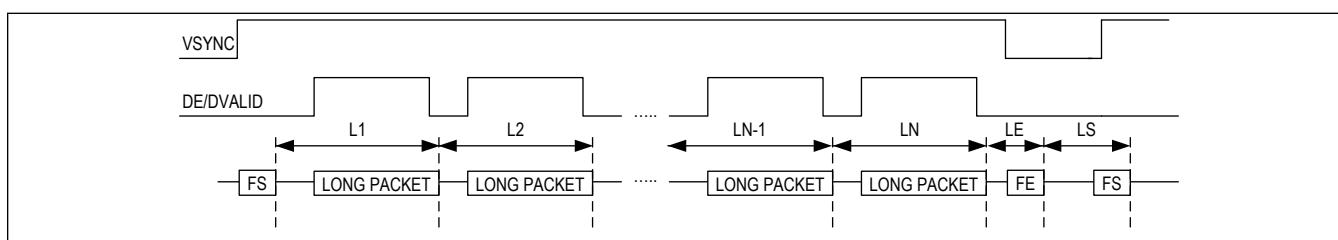


Figure 29. End-to-End Packet Spacing

### CFG Latch at Power-Up Pins

Voltage levels at the CFG0 and CFG1 pins are latched at power-up, or upon a low-to-high transition of PWDNB. These levels set initial register values and functional modes that may not be easily programmed through I<sup>2</sup>C or UART after the IC powers up. The CFG pins select device address, I<sup>2</sup>C or UART main control channel, GMSL forward data rate, and Coax or STP cable. See [Table 11](#) and [Table 12](#).

The voltage level for each pin is set by an external precision resistor-divider connected between  $V_{DDIO}$  and ground, or for some configurations, by a single resistor connected to  $V_{DDIO}$  or ground. [Table 11](#) and [Table 12](#) show the recommended resistor values to select each configuration. The voltage level at the CFG pins is latched approximately 1ms after all MAX96717 supplies reach the minimum levels required by the Power-on-Reset (POR) circuit.

If the requirements described in [Table 11](#) and [Table 12](#) are met, the CFG pins can be used as general-purpose or MFP function outputs, after the input voltage levels are latched. CFG pins cannot be used as general-purpose inputs.

**Table 11. CFG0 Input Map**

CFG0 INPUT VOLTAGE SPECIFICATION (% OF $V_{DDIO}$ ) (NOTES a, b)			SUGGESTED RESISTOR VALUES (1% TOLERANCE) (NOTE c)		MAPPED CONFIGURATION (NOTE c)		
MIN	TYP	MAX	R1 (Ω)	R2 (Ω)	I2CSEL	RoR/ Xtal	DEVICE ADDRESS
0.0%	0.0%	11.7%	OPEN	10k	I <sup>2</sup> C	RoR	0x80
16.9%	20.2%	23.6%	80.6k	20.5k			0x84
28.8%	32.1%	35.5%	68.1k	32.4k		Xtal	0x80
40.7%	44.0%	47.4%	56.2k	44.2k			0x84
52.6%	56.0%	59.3%	44.2k	56.2k		RoR	0x84
64.5%	67.9%	71.2%	32.4k	68.1k			0x80
76.4%	79.8%	83.1%	20.5k	80.6k		Xtal	0x84
88.3%	100%	100%	10k	OPEN			0x80

#### Notes

- Resistor-divider tolerance,  $V_{DDIO}$  supply ripple, and external loading must not cause the CFG0 input voltage to exceed the maximum or minimum limits.
- Other than the CFG0 input resistor-divider, any load on CFG0 must be  $\geq 25 \times (R1 + R2)$ . Each resistor in the voltage-divider must be  $\leq 100\text{k}\Omega$ .
- I2CSEL: I<sup>2</sup>C or UART interface for SDA\_RX and SCL\_TX.

**Table 12. CFG1 Input Map**

SPECIFICATION (% of $V_{DDIO}$ ) (NOTES a, b)			SUGGESTED RESISTOR VALUES (1% TOLERANCE)		MAPPED CONFIGURATION (NOTE c)		
MIN	TYP	MAX	R1 (Ω)	R2 (Ω)	COAX or TWISTED PAIR	DATA RATE (Gbps)	TUNNEL/PIXEL MODE
0%	0.0%	11.7%	OPEN	10k	STP	3	Tunnel
16.9%	20.2%	23.6%	80.6k	20.5k		6	
28.8%	32.1%	35.5%	68.1k	32.4k		3	Pixel
40.7%	44.0%	47.4%	56.2k	44.2k		6	
52.6%	56.0%	59.3%	44.2k	56.2k	COAX	3	Tunnel
64.5%	67.9%	71.2%	32.4k	68.1k		6	
76.4%	79.8%	83.1%	20.5k	80.6k		3	Pixel
88.3%	100%	100%	10k	OPEN		6	

#### Notes

- Resistor-divider tolerance,  $V_{DDIO}$  supply ripple, and external loading must not cause the CFG1 input voltage to exceed the maximum or minimum limit.
- Other than the CFG1 input resistor-divider, any load on CFG1 must be  $\geq 25 \times (R1 + R2)$ . Each resistor in the voltage-divider must be  $\leq 100\text{k}\Omega$ .

c. GMSL2: GMSL2 operating mode.

### Multifunction Pin Assignments

Side-channel functions, such as SPI and pass-through I<sup>2</sup>C/UART, are enabled by programming multifunction pins. Each MFP has several possible functions, but only one can be used at a time.

Some functions require only a single MFP, but most are implemented across a group of MFPs. For example, LOCK is a single MFP, but SPI takes several MFPs. MFP functions are selected to suit each use case by programming the appropriate registers.

The [Pin Description](#) table shows default and alternate functions for each MFP, listed in order of priority (highest first). [Table 13](#) also shows priority, left to right, with highest priority on the left. A higher-priority function must be disabled when a lower-priority function is to be enabled, both by register writes.

Each MFP defaults to one of four slew rates, with each setting having a default output transition-time setting. The transition-time default suits the MFP's normal application requirements. Except for I<sup>2</sup>C/UART and GPIO\_O/DO functions, whose slew rates are fixed, the transition time of each setting can be changed from the default value through register programming. When the slew rate is changed, the transition time of the MFP is changed. The main channel and pass-through I<sup>2</sup>C/UART channel's transition times are not affected by the transition-time settings.

Transition times depend on the transition-time setting and V<sub>DDIO</sub> supply voltage. See [Table 14](#) for typical transition times.

**Table 13. MFP Pin Function Map**

PIN	LATCH ON POWER-UP	I <sup>2</sup> C/UART	SPI	OTHER FUNCTIONS	GPIO	POWER_UP DEFAULT	PIN SLEW DEFAULT (BINARY)
MFP0			SCLK		GPIO0	DISABLED (1MΩ to GND)	10
MFP1	CFG0		BNE, SS1		GPO1	DISABLED (Hi-Z)	11
MFP2	CFG1			RCLKOUT(Alt), DPLL_OUT (Alt)	GPO2	DISABLED (Hi-Z)	11
MFP3				ERRB, LOCK, ADC0	GPIO3	DISABLED (1MΩ to GND)	11
MFP4			RO, SS2	RCLKOUT, DPLL_OUT	GPIO4	DISABLED (1MΩ to GND)	11
MFP5				LMN0, ADC1	ODO5_GPI5	DISABLED (1MΩ to GND)	NA
MFP6				LMN1, ADC2	ODO6_GPI6	DISABLED (1MΩ to GND)	NA
MFP7		SDA1_RX1	MOSI	LOCK(Alt)	GPIO07	GPI7 (1MΩ to GND, Transmit the GPIO value across the GMSL Link)	11
MFP8		SCL1_TX1	MISO	ERRB(Alt), MS	GPIO08	GPIO8 (Pulled High, Output Driver Enabled, and Ready to Receive and Output the GPIO Value from the GMSL Link)	11
MFP9		SDA_RX		SDA2_RX2	ODO9_GPIO9	SDA or RX (I <sup>2</sup> C or UART, 40kΩ to V <sub>DDIO</sub> )	NA
MFP10		SCL_TX		SCL2_TX2	ODO10_GPIO10	SCL or TX (I <sup>2</sup> C or UART, 40kΩ to V <sub>DDIO</sub> )	NA

All MFP pins are Hi-Z in power-down state.

All MFP IOs are latched in Sleep mode.

Hi-Z – 1MΩ pullup/pulldown is disabled and input/output is disabled.

Latched – Value of input before Sleep mode entered remains latched in Sleep mode and after return to Power-up mode.

ODO = Open-drain output.

**Table 14. Control- and Side-Channel Typical Rise and Fall Times**

PIN SLEW	RISE TIME (ns) (20% to 80%), $C_L = 10\text{pF}$		FALL TIME (ns) (80% to 20%), $C_L = 10\text{pF}$	
	$V_{DDIO} = 1.8\text{V}$		$V_{DDIO} = 3.3\text{V}$	
			$V_{DDIO} = 1.8\text{V}$	$V_{DDIO} = 3.3\text{V}$
00	1.0	0.6	0.8	0.5
01	2.1	1.1	2.0	1.1
10	4.0	2.3	4.3	2.3
11	9.0	5.0	10.0	5.0
I <sup>2</sup> C	N/A	N/A	40	30

### Control-Channel Link and Power-Up

GMSL2 ICs are in power-down mode when the PWDNB pin is low or when any of the power supplies are down. Register and configurations are set to default reset conditions.

The serializer and deserializer may power up in any order. After all power supplies are up and PWDNB is released, each device starts its power-up sequence and performs these actions in sequence:

1. Latch at power-up pins register set. Set internal registers according to the selected configuration on CFG0 and CFG1 pins. See [Table 11](#) and [Table 12](#).
2. Control channel (I<sup>2</sup>C or UART) is functional on the local side. Device registers are writable and readable.
3. The enabled PHY performs link calibration, equalizer adaptation, and data-channel locking. Once the link is locked, the device sets the LOCK pin high.
4. Control channel is available from the remote side.

This entire link-up process, from the time that the last part's PWDNB input is brought high, takes approximately 45ms nominally for any channels that meet the GMSL2 channel specification.

After the device is linked, it can be configured. This can be done either locally or over the control channel by a μC on either the serializer or deserializer side.

### Device Reset

There are three general reset options available through register writes:

- RESET\_ALL resets all blocks, including all registers, digital, and analog blocks. This is similar to driving the PWDNB pin low and then high. Note: If Sleep mode is being used, do not use RESET\_ALL as it returns the device to Sleep mode.
- Setting RESET\_LINK resets all GMSL2 PHY-related digital logic and all data pipelines. After this bit is set, all control registers are still accessible through the local control channel. The link remains in RESET until RESET\_LINK is cleared.
- RESET\_ONESHOT resets all GMSL2 PHY-related digital logic and all data pipelines, and then automatically clears itself. This is similar to setting and clearing RESET\_LINK. (Note: For the purposes of achieving predictable and expected link lock time, it is recommended that RESET\_ONESHOT is not used while link lock operation is in progress. This includes the initial link lock attempt immediately following power-up. If link lock operation must be interrupted, RESET\_LINK should be used instead to hold the link in reset until both SER and DES are ready to establish link lock. RESET\_ONESHOT can be used any time after the link has been locked to reset and predictably relock the link.)

Registers that affect GMSL2 link operation (i.e., TX\_RATE, RX\_RATE, CXTP\_A) should be programmed first, followed by RESET\_ONESHOT. Alternatively, set these registers when RESET\_LINK = 1, and then set RESET\_LINK = 0. Always disable the UART pass-through channel before resetting the link.

### Clocking

## GMSL Reference Clock

The MAX96717 requires a reference clock source to generate the 6GHz line-rate clock and associated internal clocks. There are two clocking modes, Crystal mode or Reference over Reverse (RoR) mode. Clocking mode is determined at start-up from the resistors connected to the CFG0 pin. In the crystal mode, both the serializer and deserializer can be clocked with an external 25MHz crystal or an external clock source with a frequency accuracy of  $\pm 200\text{ppm}$ . In the RoR mode, no crystal is needed for the serializer. The MAX96717 generates a reference clock using information embedded in the return-path data-stream with an integrated digital phase-locked loop. As the serializer also can generate the reference clock for an image sensor and other components in a sensor module, RoR allows elimination of crystals from sensor modules. This also phase and frequency locks sensor modules to a common frequency reference, which is connected to the deserializer. See the [Reference Over Reverse Channel](#) section.

## Reference Clock Input Jitter

If the reference clock is supplied from an external clock source rather than from a crystal, reference clock jitter must be considered. Jitter tolerance is highest for slow variations in clock period, as these can be absorbed by the clock recovery circuit on the GMSL2 serial link.

## Remote Clocking (RoR Channel)

### Reference Over Reverse/RoR Channel

The MAX96717 supports the generation of the reference clock for the device using the reverse-channel data stream. This eliminates the need for a local 25MHz crystal in each sensor. A phase-locked loop is used to synchronize the serializer's reference clock to the reference clock in the deserializer. This eliminates the need for a local crystal and synchronizes the timing for all devices using the serializer's reference clock output.

### RoR Power-Up

At power-up and when the MAX96717 is configured to operate in RoR using the configuration pins, a free-running oscillator within the serializer is initially used as a clock reference until the RoR signal being sent from the deserializer to the serializer's return-path receiver is detected. Once the return-path signal's embedded reference locks the serializer's PLL, a locked state signal is sent from the serializer to the deserializer on the serializer's forward path as part of the power-up sequence. After the deserializer receives the lock signal from the serializer, control-channel data is enabled over the reverse channel, and the serializer's forward channel is enabled. The serializer is then ready for register programming and normal operation. Note that the deserializer must support the RoR mode.

### Reference Clock/DPLL Generation

The MAX96717 can share a crystal or reference by providing a reference clock output. RCLKOUT can be used as a reference clock by a sensor or other IC in close proximity, eliminating the need for an additional external crystal or oscillator in a camera module.

The MAX96717 can also generate a unique reference frequency by utilizing an integrated digital PLL, which can be configured to generate a multitude of output frequencies through DPLL\_OUT. The reference clock is output on the MFP4 pin.

Using the built-in reference clock divider, the part can supply 25MHz, 12.5MHz, or 6.25MHz. Using the digital PLL allows frequency generation over a frequency range of 1MHz to 75MHz with fractional step sizes. Predefined frequencies include 13.5MHz, 19.2MHz, 24MHz, 27.0MHz, 37.125MHz, and 74.25MHz. By configuring the dividers in the DPLL,  $f_{\text{DPLL\_OUT}} = (N \cdot M) / K \times f_{\text{REF}}$ , where N, M, and K are 7-bit, 11-bit, and 9-bit integers generated, respectively. Refer to the GMSL2 User Guide for further details on programming the digital DPLL.

### Spread-Spectrum Clocking

Analog Devices' GMSL2 links provide exceptional EMI performance. Optional spread-spectrum clocking (SSC) is available to further mitigate EMI caused by the emissions. SSC reduces peaks in the frequency spectrum by spreading the signal over a wider bandwidth. The spread has a 25kHz sawtooth modulation profile, programmable to deviate up to  $\pm 1265\text{ppm}$  from the center frequency.

## Error and Fault-Condition Monitoring

Both the serializer and deserializer have an open-drain, multipurpose error reporting and interrupt status output. The active-low ERRB pin is driven by the logical OR of a wide variety of error and event status indicators. The ability of each error condition to drive ERRB is maskable by register settings. Each error and event that can drive ERRB has a status flag within a sub-block of registers. So, the reason for assertion of ERRB can be determined by reading the register status.

When relying on the ERRB pin to convey the occurrence of an undervoltage event, connect an external  $1\text{M}\Omega$  resistor between the ERRB pin and the ground, because ERRB is not a power-up default MFP function. As a result, following a reset that is triggered by an undervoltage event, the ERRB MFP pin transitions through high-impedance states. During an error state, the host device expects ERRB to drive logic-low, and the presence of the external  $1\text{M}\Omega$  resistor enables the appropriate logic level to be maintained following the reset, alerting the host device that attention is required.

## Power Supplies

The supply voltages for the MAX96717 can be brought up in any order. An on-chip power manager ensures that all supply rails are within limits before enabling device start-up.

The serializer core runs on a regulated 1.0V supply (CAP\_VDD). It is supplied from a built-in LDO that regulates the voltage received on the  $V_{DD}$  pin down to 1.0V. CAP\_VDD also powers the output driver for the GMSL forward channel.

The  $V_{DDIO}$  supply for the GPIO pins can be between 1.8V and 3.3V for flexibility in interfacing with the part. The allowable supply voltage range is 1.7V to 3.6V.

$V_{DD18}$  is the analog supply. Connect to 1.8V.

Proper bypassing of all power supplies is essential for high-frequency circuit performance. See [Table 3](#) and [Table 2](#) for power-supply tolerances and noise requirements. Contact the factory for guidance on sharing supplies and optimizing supply decoupling.

Analog Devices provides power management ICs (PMICs) optimized for supporting serial link devices. Contact the factory for information.

## Supply Sequencing

The power supplies to the MAX96717 can be brought up in any sequence. An on-chip power management block manages the power domains during power-up.

However, when a PMIC has the ability to sequence, the following sequence is recommended:

1.  $V_{DD18}$  ramp-up
2.  $V_{DDIO}$  ramp-up
3.  $V_{DD}$  ramp-up

Power supply ramp-time recommendation:  $20\mu\text{s} < \text{ramp time} < 2\text{ms}$ . Power supply ramps should be monotonic. Once the supply voltage reaches the minimum supply voltage limit, it should not be allowed to drop below the specification.

## Overvoltage/Undervoltage

The MAX96717 includes overvoltage (OV) and undervoltage (UV) comparators and detection logic for the supply rails.

For an OV condition on  $V_{DD18}$ , CAP\_VDD (internal 1V), or  $V_{DD}$ , the ERRB pin is asserted, but no further action is taken. The device continues to operate and may suffer damage. Note that OV detection is active only when the pixel clock detector (PCLKDET) detects a valid pixel clock. To clear the OV flags, read bits VREG\_OV\_FLAG, VDD\_OV\_FLAG, and VDD18\_OV\_FLAG.

In a UV condition, the ERRB pin is asserted when:

1.  $V_{DD18}$  drops below an internal threshold. The part continues to function until  $V_{DD18}$  drops further to the reset level, and the device resets.
2.  $V_{DDIO}$  drops below an internal threshold. The part continues to function until  $V_{DDIO}$  drops further to the reset level, and the device resets. There are no OV flags for the  $V_{DDIO}$  rail.
3. CAP\_VDD (core 1V supply) drops below an internal threshold. The device resets.

Once the device resets and the power supply recovers, the device returns to the default power-up state with default

register settings. Reinitiate device start-up and reprogram the device.

To clear the UV flags: read the respective error status registers PWR0 and PWR1 and the respective flags (VDDBAD\_INT\_FLAG, PORZ\_INT\_FLAG, and VDDCMP\_INT\_FLAG) for each supply domain. If VDDCMP\_INT\_FLAG and PORZ\_INT\_FLAG are both set, read PWR0 before PWR1 to properly clear the ERRB pin.

When an undervoltage event occurs, I<sup>2</sup>C communication may be lost. When power is restored, UV status registers can be read to determine which power supply experienced the undervoltage event.

## PCB Layout Guidelines for GMSL

Proper circuit board design techniques are required for optimal GMSL link performance. This includes having at least a four-layer board to provide a proper ground plane reference, low thermal impedance, DC supply pin bypassing, PoC design, and high-speed GMSL trace impedance matching.

### Ground Plane

A consistent ground plane, generally the second layer, is recommended to provide isolation from the high-speed GMSL traces and other traces and components that can couple noise onto the GMSL signals. This also simplifies the design of impedance-matched transmission lines. The ground plane is also key to provide a low thermal impedance to the device's exposed paddle.

### High-Speed GMSL Traces

Proper transmission-line design techniques are needed to achieve optimal GMSL link performance. The characteristic impedance must be closely matched to the desired impedance (50Ω single-ended or 100Ω differential). Any mismatch increases insertion loss and causes reflections (degrade return loss). Suggested layout practices are:

- Use only 100Ω differential or 50Ω single-ended traces.
- Minimize length of high-speed traces.
- Minimize pad stubs by placing component pads directly on the signal trace.
- Use ground cutouts under pads as required (1.3 x area of pad).
- Follow vendor layout recommendations for components, including connectors.
- Avoid sharp bends on high-speed traces.
- Place AC-coupling capacitors within 500mils of the SIOx pins.
- Place the GMSL device as close to the serial-link connector as possible to minimize insertion loss.
- Stitch ground layers together with vias near high-speed traces.

### Power-over-Coax Layout

When used, the PoC circuit needs to provide a low DC series resistance to the power supply while minimizing the loading of the GMSL forward and reverse signals. The PoC circuit behaves as a bias-tee and requires proper layout techniques for optimal GMSL link performance, including:

- Placing the smallest valued inductor (highest self resonant frequency) on the GMSL signal trace.
- Placing the next smallest valued inductor after the smallest.
- Using ground cutouts as needed under the first two inductors and resistors. See [Figure 30](#).
- Placing the PoC within 1/2 UI of the device, particularly for the deserializer.

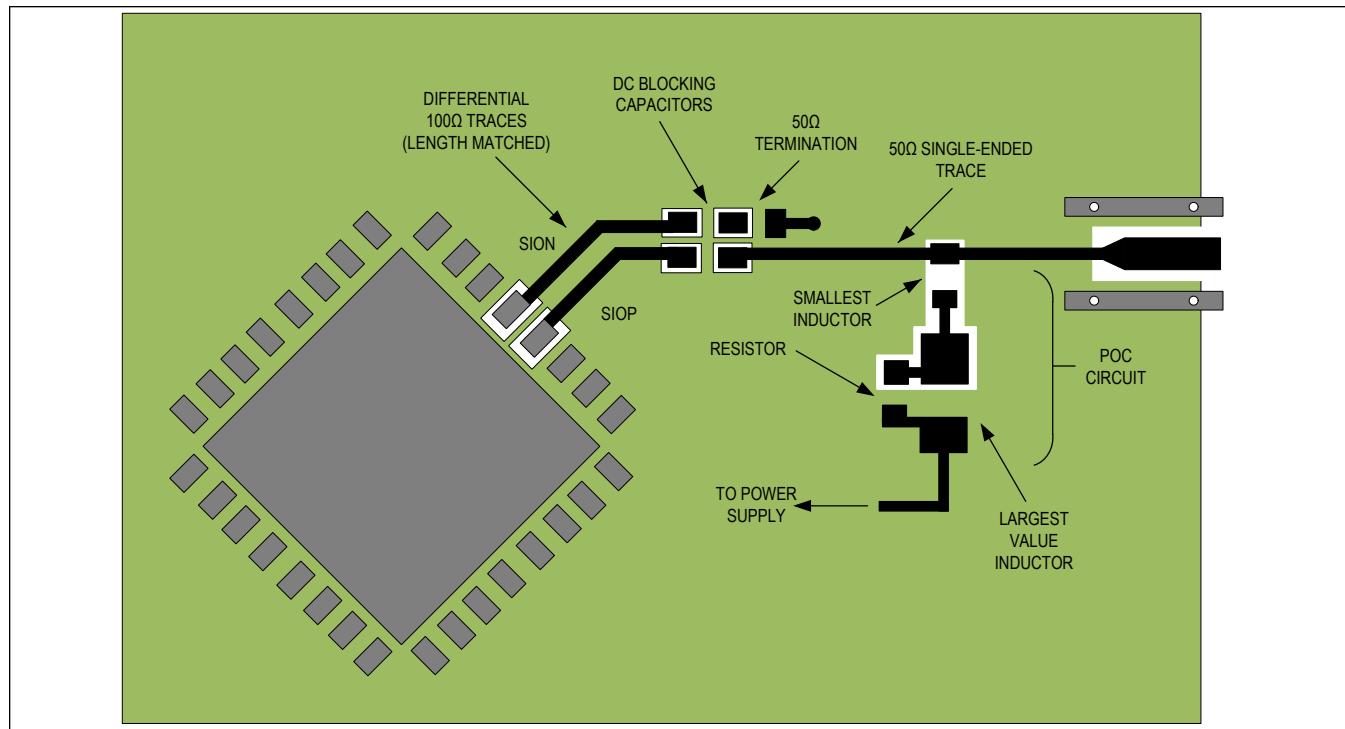


Figure 30. Coax PoC Layout Example

### Shielded-Twisted Pair Layout

Shielded-Twisted Pair (STP) designs require the differential traces to closely maintain a differential impedance of  $100\Omega$ . Use ground cutouts under AC-coupling capacitors and line-fault resistors. See [Figure 31](#).

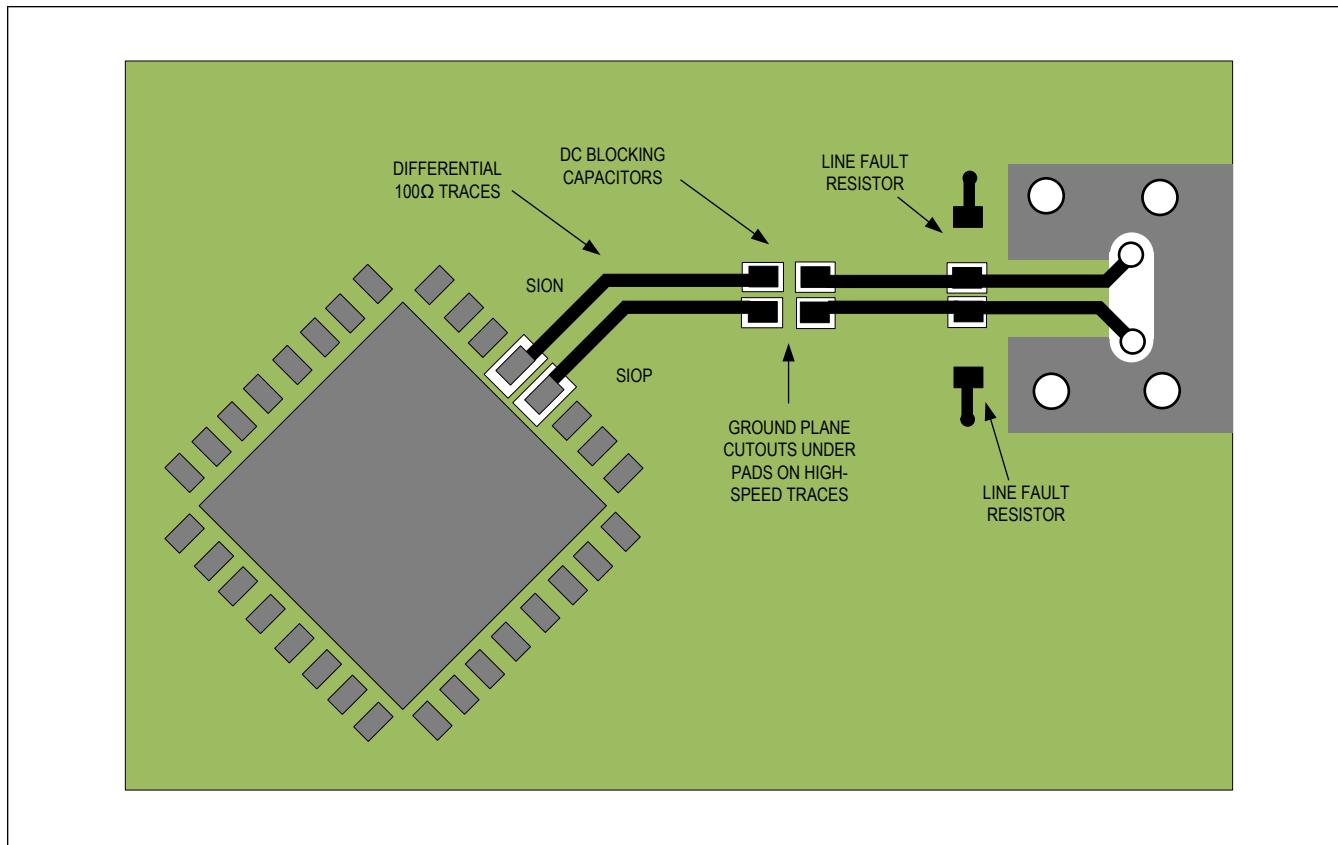


Figure 31. STP Layout Example

### Thermal Management

Power consumption of GMSL2 devices varies based on use case. The user must take care to provide sufficient heat dissipation with proper board and cooling design techniques. The exposed pad of the package must be connected to the PCB ground plane by an array of vias. This approach simultaneously provides the lowest electrical and thermal impedances.

System thermal management must keep the operating junction temperature below +125°C to avoid impacting device reliability.

Refer to [Thermal Characterization of IC Packages](#) for further guidance.

### Unique Serial Number

The device has a set of registers that are programmed at the factory with a device level unique serial number (USN).

The USN can be read to identify and validate camera modules. In addition, the USN can be used as part of an authentication scheme to prevent counterfeit/malicious devices. The numbers are not sequential but are guaranteed to be unique.

Registers 0x1C50-0x1C5E contain the serial number.

Registers 0x1C5F and 0x1C60 contain a 16-bit CRC of the serial number.

These registers can only be read from the deserializer side and are not available through local serializer I<sup>2</sup>C/UART pins.

## Applications Information

### Software Programming Model

Analog Devices' automotive serializers and deserializers are designed to follow a general software programming model. Except for features that require in-operation control channel accesses, such as the ASIL safety measures and interrupt handling, use the following programming model:

1. Set the impacted functional blocks to disabled or reset mode. A general method used to place the part in IDLE state is to stop all side-channel and video traffic, followed by a register write (RESET\_LINK = 1) to stop the GMSL link.
2. The settings for each feature must be fully configured before it is enabled.
3. Establish the link by setting RESET\_LINK = 0. Wait for the link to lock.
4. Start video and side-channel traffic.

If changing the configuration of a feature is required during the operation of other features, disable the feature that will be reconfigured, change its settings, and reenable it.

### Programming Notes

This device does not require setting bits [6:4] in register 0x302. Legacy systems that are writing [6:4] = 3'b001 in register 0x302 are still valid and operation is guaranteed.

### Control-Channel Programming

At power-up, GMSL2 device registers are accessed and configured only through the main I<sup>2</sup>C/UART interface. By default, the main I<sup>2</sup>C/UART channel is also sent to the remote side device and any peripheral connections. For multimain configurations, with microcontrollers connected to both the serializer and deserializer, disabling the remote control channel through register settings is recommended to prevent bus contention.

### Conventional I<sup>2</sup>C/UART Control-Channel Programming

#### Host-to-Peripheral Main I<sup>2</sup>C and Pass-Through I<sup>2</sup>C Communication

When communicating between a host and peripheral, main and pass-through I<sup>2</sup>C operations are the same. A pass-through I<sup>2</sup>C across the GMSL2 link connects the host's I<sup>2</sup>C main to the peripheral's I<sup>2</sup>C subordinate. This logically connects separated I<sup>2</sup>C buses, enabling I<sup>2</sup>C transactions across the serial link to occur (with some delay) as if performed on the same physical I<sup>2</sup>C bus. The GMSL2 serializer and deserializer are intermediary devices; the host I<sup>2</sup>C main connects to a GMSL2 device I<sup>2</sup>C subordinate, and the peripheral I<sup>2</sup>C subordinate connects to a GMSL2 device I<sup>2</sup>C main.

For example, when the host I<sup>2</sup>C main transacts on one side of the link (local side), data is forwarded to the other side (remote side) by the I<sup>2</sup>C subordinate of the local side GMSL2 device. Data is then received by the I<sup>2</sup>C main of the remote side GMSL2 device, which generates the same I<sup>2</sup>C transaction with the peripheral subordinate I<sup>2</sup>C. The remote side GMSL2 device sends back any I<sup>2</sup>C data expected by the local side.

**Note:** This device does not support pass-through I<sup>2</sup>C/UART channels to access the main I<sup>2</sup>C/UART control channel on the remote side.

The I<sup>2</sup>C interface uses clock stretching (holding SCL low) to account for timing differences between the main and subordinate, and to allow time for data to be forwarded and received across the serial link. All local side I<sup>2</sup>C devices must support clock stretching by the GMSL2 device. Remote side I<sup>2</sup>C devices are not required to support clock stretching.

SDA and SCL lines operate as both input and open-drain output. Pullup resistors are required on SDA and SCL.

Each transmission consists of a START condition sent by a main, followed by the device's 7-bit subordinate address plus a R/W bit, register address bytes, one or more data bytes, and finally a STOP condition.

Register addresses are 16 bits wide. Single or multiple data bytes can be written or read by address auto-increments.

### I<sup>2</sup>C Write Packet Format

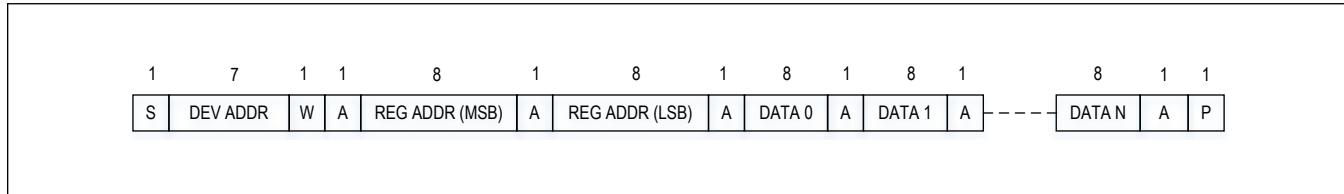


Figure 32. I<sup>2</sup>C Write Packet Format

### I<sup>2</sup>C Read Packet Format

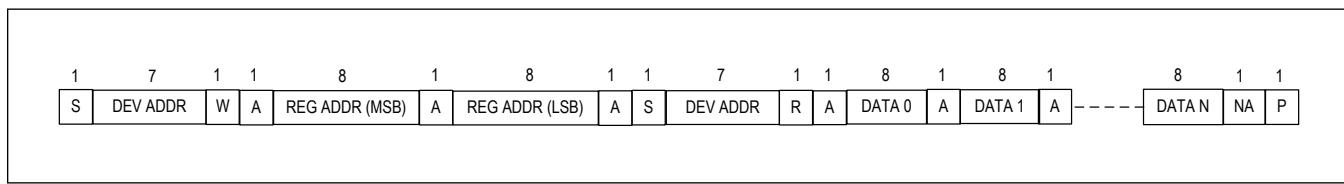


Figure 33. I<sup>2</sup>C Read Packet Format

### Main I<sup>2</sup>C Host-to-GMSL2 Device Communication

The host I<sup>2</sup>C main has access to GMSL2 serializer and deserializer registers. The host can program GMSL2 device registers to configure the pass-through I<sup>2</sup>C/UART interface as I<sup>2</sup>C or UART.

### Main UART

When the main I<sup>2</sup>C/UART is configured as a UART, there are two operating modes: Base and Bypass modes.

#### UART Base Mode

Base mode is the means by which the microcontroller communicates with the serializer and deserializer, where registers in these and peripheral devices can be accessed. Base mode is typically enabled by default at power-up. In Base mode, the  $\mu$ C is the host and can access the registers of both the serializer and deserializer from either side of the link using the GMSL2 UART packet protocol. The  $\mu$ C can also program the peripherals on the remote side by sending the UART packets to the serializer or deserializer. The  $\mu$ C communicates with a UART peripheral in Base mode (through INTTYPE register settings). The device addresses of the serializer and deserializer in this mode are programmable. In Base mode, the serializer, deserializer, and peripheral registers can be written and read using the half-duplex GMSL2 UART protocol. Base mode is enabled by default at power-up.

[Figure 34](#) shows the UART protocol for writing and reading in Base mode.

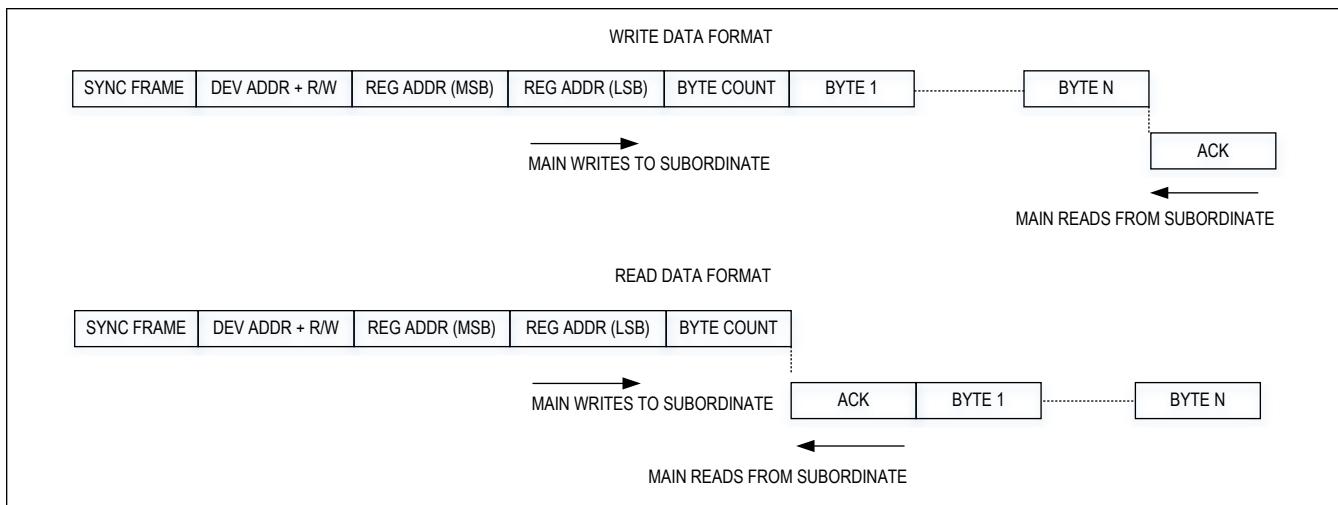


Figure 34. UART Protocol for Base Mode

### UART Bypass Mode

In the bypass mode, the serializer/deserializer ignore UART commands from the  $\mu$ C, and the  $\mu$ C communicates only with the peripherals using its own defined UART protocol. The  $\mu$ C cannot access the serializer/deserializer's registers in this mode. The UART transitions are simply sent over the GMSL2 link. Ignoring UART transactions prevents inadvertent misprogramming of serializer and deserializer registers. The device addresses of the serializer and deserializer in this mode are not programmable.

### Switching Between UART Base and Bypass Modes

There are two ways to switch between the base and bypass modes: programming the BYPASS\_TO register and using the device MS pin.

When setting the bypass mode through the register, BYPASS\_TO is programmed for a timeout (2ms, 8ms, or 32ms). Bypass mode is active only as long as there is UART activity. When there is no UART activity for the selected timeout, both devices exit the bypass mode, and the bit is automatically cleared.

When in the UART bypass mode, random data may be output if link lock is lost. This issue can be avoided by not enabling the UART bypass mode until all initial device programming is complete.

When set by the MS pin, a high-level puts the device into the bypass mode, and a low-level puts the device into the base mode. MS is set on the fly and is not latched on power-up.

### UART Frame Format

Regular UART frames with an even parity bit are used to carry 1 byte of data each. A frame consists of a low start bit followed by 8 data bits, a parity bit, and a high stop bit. The parity bit is high if the number of 1s in 8-bit data is odd; otherwise, it is low. There must be at least 1 high stop bit. If the next frame is in the same packet, there can be no more than 4 high bits from the end of the stop bit to the beginning of the next start bit. Note that for a parity-bit error, the packet, starting from the frame with the error, is discarded. The start of each frame is always a high-to-low transition (i.e., the stop bit is high and start bit is low). The phase of the internal UART bit clock is adjusted using the start bit of each frame. The framer calibrates the length of 1 UART bit in terms of the internal oscillator clock using the synchronization frame (i.e., the first frame of a UART packet transmission). In Bypass mode, the parity bit is enabled by default, but the frames are not checked for parity errors. Either even or odd parity can be used. The parity bit is passed along with UART data transmissions; the recipient of the data must perform error checking. The parity bit can optionally be disabled before entering Bypass mode. Note that the bit rate in Bypass mode must be the same bit rate last used in Base mode. See [Figure 35](#).

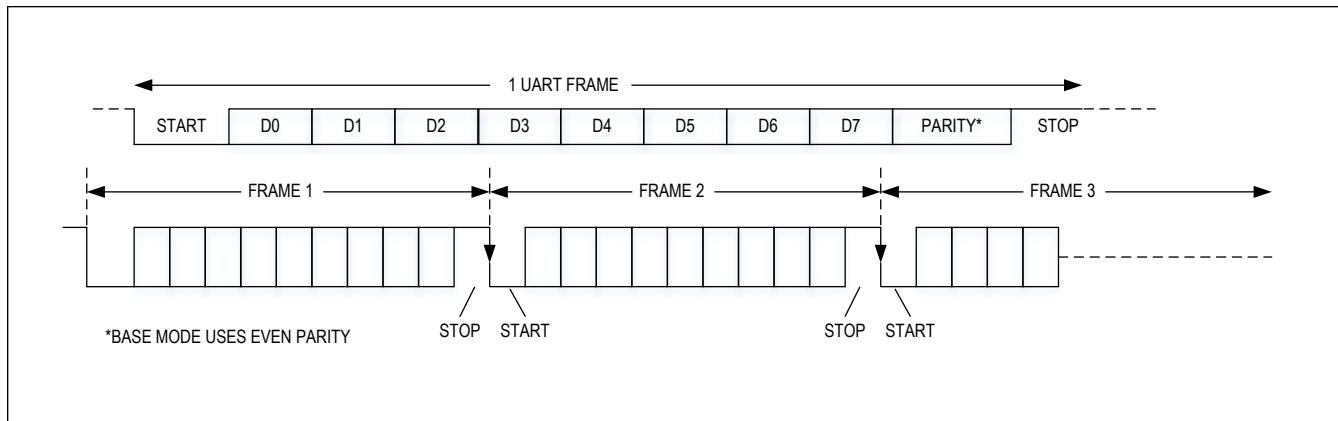


Figure 35. UART Data Format for Base Mode

### Synchronization Frame

The serializer/deserializer must calibrate internal bit length counters with the UART bit rate for the proper recovery of UART frames. The  $\mu$ C sends a synchronization frame (i.e., a regular UART frame with the value 0x79) as the first frame of each data packet. The synchronization frame allows the addressed device to calibrate the bit length in terms of the device's internal 150 MHz clock. A synchronization frame must be properly detected before the subsequent frames of the packet can be correctly received. When the line stays high for at least 32 bits, the packet boundary is reset, and the framer begins waiting for the next synchronization frame. See [Figure 36](#).

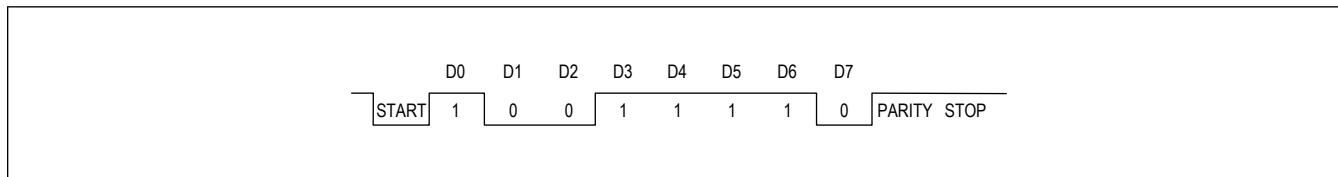


Figure 36. UART Synchronization Frame

### Acknowledge Frame

When a packet is successfully received, the addressed device responds with an acknowledge frame to inform the  $\mu$ C that no errors are detected in the transmitted packet. The acknowledge frame is sent after the last bit of a valid packet is received. The acknowledge frame is a regular UART frame (value 0xC3). Data written to the serializer/deserializer registers do not take effect until after the acknowledge byte is sent. See [Figure 37](#).

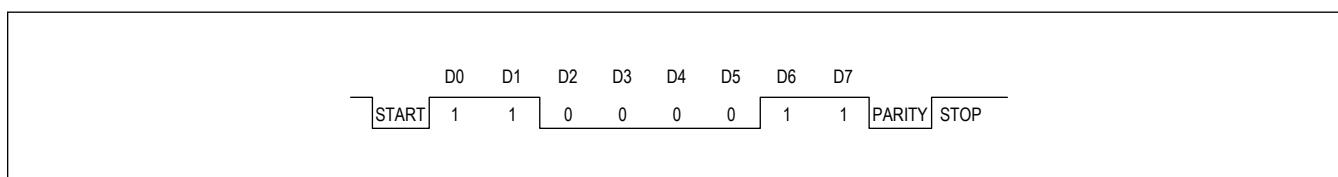


Figure 37. UART Acknowledge Frame

### Write Packet

Write packets consist of a 5-byte packet header followed by 1 or more data bytes. A packet is recognized as a write packet when the LSB of the device address frame is 0. The addressed device responds with an acknowledge frame if no errors are detected while receiving a valid write packet. Byte Count indicates the number of data bytes to be written; this number cannot be 0. See [Figure 38](#).

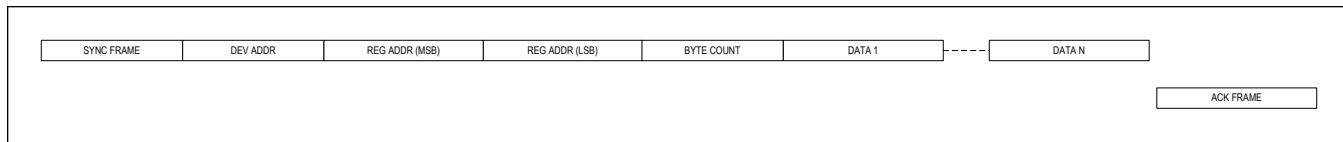


Figure 38. UART Write Packet Format

### Read Packet

Read packets consist of 5 bytes. The LSB of the device address frame is 1 for read packets. If no errors are detected while receiving a valid read packet, the addressed device responds with an acknowledge frame followed by 1 or more data bytes. Byte Count indicates the number of data bytes to be read; this number cannot be 0. See [Figure 39](#).

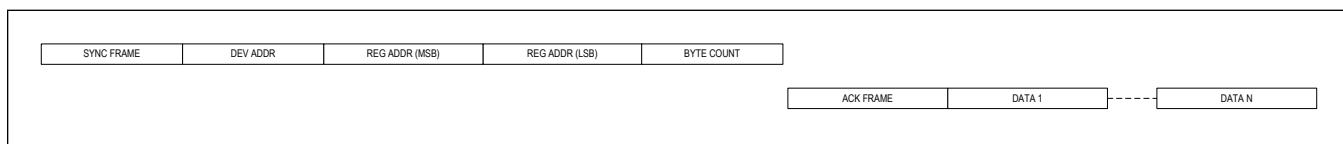


Figure 39. UART Read Packet Format

### I<sup>2</sup>C/UART Control-Channel Programming with Optional CRC and Message Counter

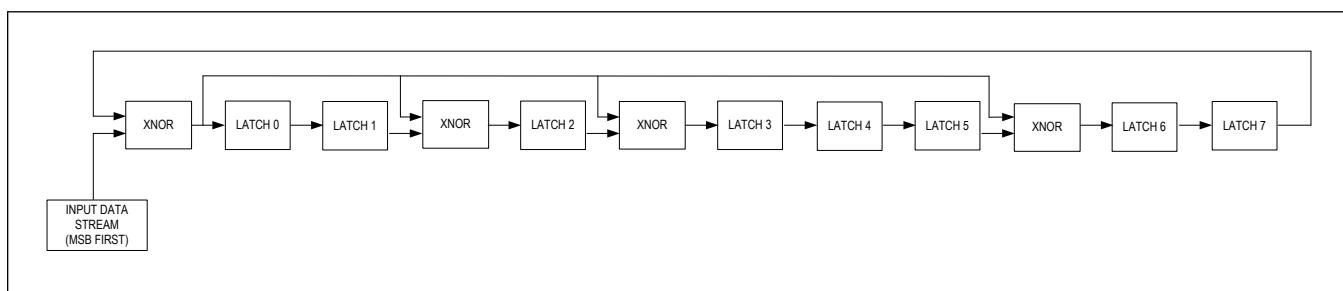
The MAX96717 provides additional functional safety by adding an optional CRC to I<sup>2</sup>C/UART read/write transactions, and a separate message counter for read and write packets. These features are disabled by default at power-up, but can be enabled by register settings. The CRC and message counter can be used together or individually. The CRC and message counter features only apply to device register read/write transactions, and are not supported for pass-through traffic. The CRC and/or message counter can be enabled in the serializer, deserializer, or both.

### I<sup>2</sup>C/UART CRC and Message Counter Options

At power-up, the I<sup>2</sup>C/UART CRC and Message Counter features are disabled by default in the MAX96717. Both features can be enabled or disabled through register control.

### I<sup>2</sup>C Writes with CRC

To provide additional functional safety for advanced driver assistance systems (ADAS) applications, the MAX96717 supports the addition of a cyclic redundancy check (CRC) to I<sup>2</sup>C transactions. When enabled, the main µC must compute and send a CRC byte after each data byte. See [Figure 40](#). For each single-byte or multibyte write, the serializer first clears the CRC engine. The first CRC includes the device address byte, register address bytes, message counter bytes, and first data byte. For all following data bytes, the CRC engine is reset, and the CRC byte covers the additional data byte. See [Figure 41](#) and [Figure 42](#). The serializer receives the data byte and calculates the CRC using an identical CRC engine. It verifies a match before accepting the data byte. If the CRCs do not match, a write is not accepted, a NACK is transmitted, and the error counter is triggered.

Figure 40. I<sup>2</sup>C CRC Engine

1	7	1	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	1	
S	DEV ADDR	W	A	REG ADDR (MSB)	A	REG ADDR (LSB)	A	MSG COUNTER (MSB)	A	MSG COUNTER (LSB)	A	DATA 0	A	CRC	A	DATA 1	A	CRC	A	---	DATA N	A	CRC	A	P

Figure 41. I<sup>2</sup>C Multiple-Byte Write with CRC

1	7	1	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	1	1		
S	DEV ADDR	W	A	REG ADDR (MSB)	A	REG ADDR (LSB)	A	MSG COUNTER (MSB)	A	MSG COUNTER (LSB)	A	DATA 0	A	CRC	A	DATA 1	A	CRC	A	---	DATA N	A	CRC	A	P

Figure 42. I<sup>2</sup>C Single-Byte Write with CRC

## I<sup>2</sup>C Reads with CRC

For I<sup>2</sup>C reads of the MAX96717's registers, the device's CRC engine clears, then uses the outgoing device address byte, register address bytes, message counter bytes, and first data byte to calculate the CRC byte. This is added to the data stream directly after the corresponding data byte. The CRC engine is subsequently cleared again for all other data bytes. See [Figure 43](#) and [Figure 44](#). When the  $\mu$ C receives the I<sup>2</sup>C read data, either through the MAX96717 or directly from the device on the other side of the link, the  $\mu$ C's CRC engine should calculate a CRC byte for each data byte and compare with the transmitted CRC byte.

1	7	1	1	8	1	7	1	1	7	1	1	8	1	8	1	8	1	8	1	8	1	1	1						
S	DEV ADDR	W	A	REG ADDR (MSB)	A	REG ADDR (LSB)	A	S	DEV ADDR	R	A	MSG COUNTER (MSB)	A	MSG COUNTER (LSB)	A	DATA 0	A	CRC	A	DATA 1	A	CRC	A	---	DATAN	A	CRC	NA	P

Figure 43. I<sup>2</sup>C Multiple-Byte Read with CRC

1	7	1	1	8	1	7	1	1	7	1	1	8	1	8	1	8	1	8	1	8	1	1	1						
S	DEV ADDR	W	A	REG ADDR (MSB)	A	REG ADDR (LSB)	A	S	DEV ADDR	R	A	MSG COUNTER (MSB)	A	MSG COUNTER (LSB)	A	DATA 0	A	CRC	A	DATA 1	A	CRC	A	---	DATA N	A	CRC	NA	P

Figure 44. I<sup>2</sup>C Single-Byte Read with CRC

## Message Counter Writes

An additional functional safety feature of the MAX96717 is a message counter, which can be used with the I<sup>2</sup>C/UART CRC feature. If enabled, the device expects a 2-byte message count from the  $\mu$ C, indicating the number of writes being sent. The MAX96717 counts the number of write transactions and compares that with the count sent by the  $\mu$ C. If the two counts match, the write is accepted. If the two counts do not match, the write is rejected, a NACK is sent in return, and the error counter is incremented. A programmable threshold for the error counter asserts ERRB, when reached.

## Message Counter Reads

For read transactions, the MAX96717 sends the message counter value to the  $\mu$ C along with the requested data. The  $\mu$ C should compare the sent message count against its stored value and accept the data if the counts match. Note that a read transaction has a repeated start condition with the device address byte sent twice. This results in each read transaction incrementing the message counter twice, once for each device address byte. If the two message counter values do not agree, the data should be rejected.

If a read is requested from the last used registers address, resulting in only one device address byte, the message counter is incremented once.

If the message counter values for the MAX96717 and  $\mu$ C do not agree, the device's message counter can be reset using a register write; the  $\mu$ C's counter should also be reset. If the  $\mu$ C or its message counter is reset for any reason, the device's counter must be reset using a register write as well.

## UART Writes with CRC

To provide additional functional safety for ADAS applications, the MAX96717 supports the addition of a CRC to UART transactions. When enabled, the main  $\mu$ C must compute and send a CRC byte after each data byte. See [Figure 45](#). For each single-byte or multibyte write, first clear the CRC engine. The first CRC includes the synchronization frame, device address byte, register address bytes, message counter bytes, and the first data byte. All bytes are sent least significant byte (LSB) first. For all following data bytes, the CRC engine is reset, and the CRC byte covers the additional

data byte. See [Figure 46](#) and [Figure 47](#). The MAX96717 receives the data byte, calculates the CRC using an identical CRC engine, and verifies a match before accepting the data byte. If the CRCs do not match, a write is not accepted, a NACK is transmitted, and the error counter is triggered.

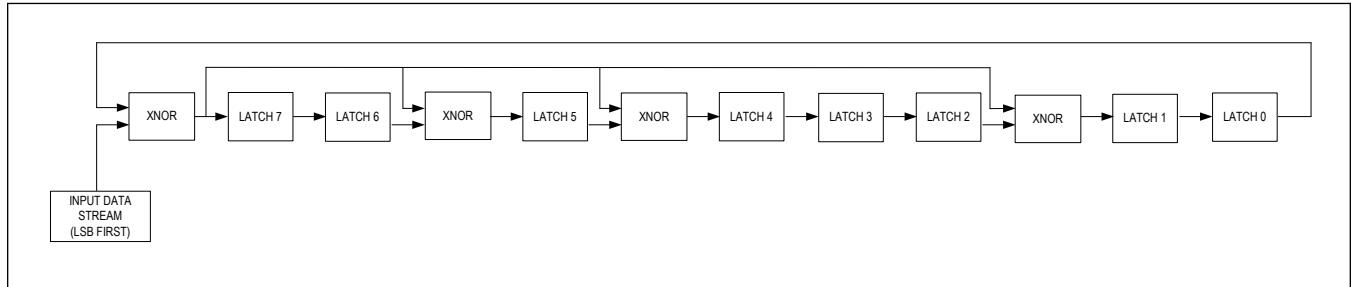


Figure 45. UART CRC Engine

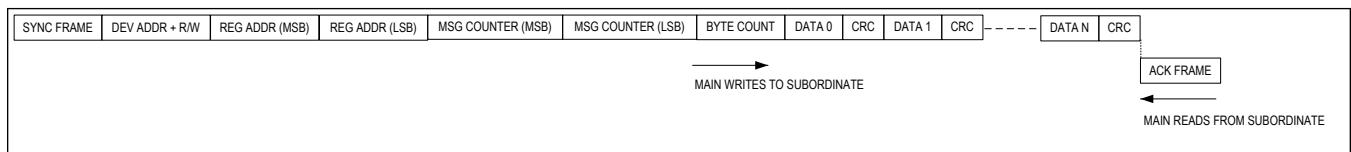


Figure 46. UART Multiple-Byte Write Transactions with CRC

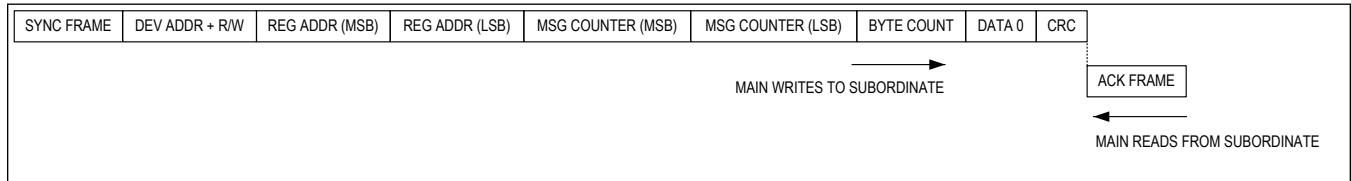


Figure 47. UART Single-Byte Write Transaction with CRC

### UART Reads with CRC

For UART reads of MAX96717's registers, the device's CRC engine clears, then uses the outgoing sync frame, device address byte, register address bytes, message counter bytes, and the first data byte to calculate the CRC byte. This is added to the data stream directly after the corresponding data byte. All bytes are sent LSB first. The CRC engine is subsequently cleared again for all other data bytes. See [Figure 48](#) and [Figure 49](#). When the  $\mu$ C receives the UART read data, either through the serializer on the other side of the link or directly from the MAX96717, the  $\mu$ C's CRC engine can calculate a CRC byte for each data byte and compare with the transmitted CRC byte. The ACK frame is not included in the CRC calculation, because the  $\mu$ C is looking for the ACK value of 0xC3. If there is an error, the  $\mu$ C treats the frame as a NACK and rejects the data. Note that the read command from the  $\mu$ C does not include CRC.

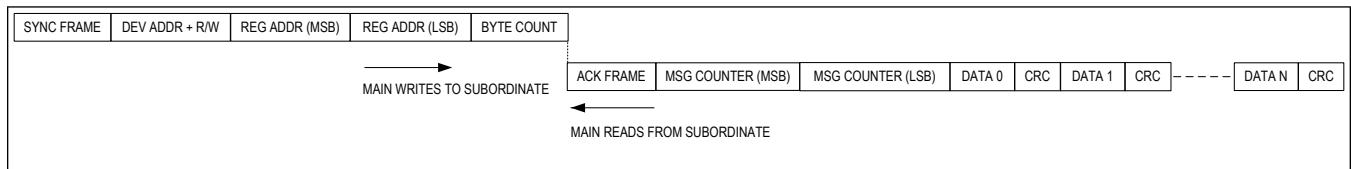


Figure 48. UART Multiple-Byte Read Transaction with CRC

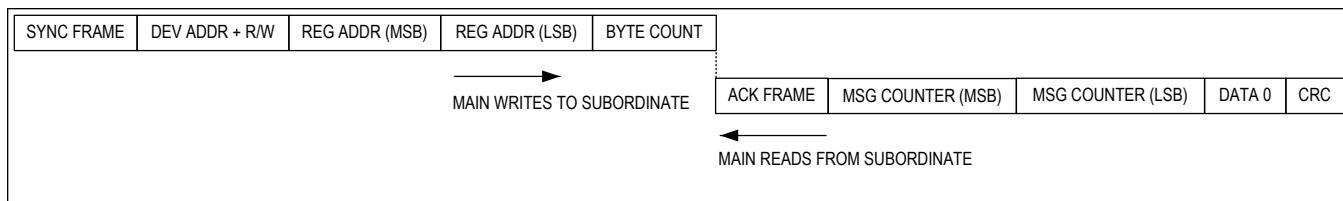


Figure 49. UART Single-Byte Read Transaction with CRC

## Typical Application Circuits

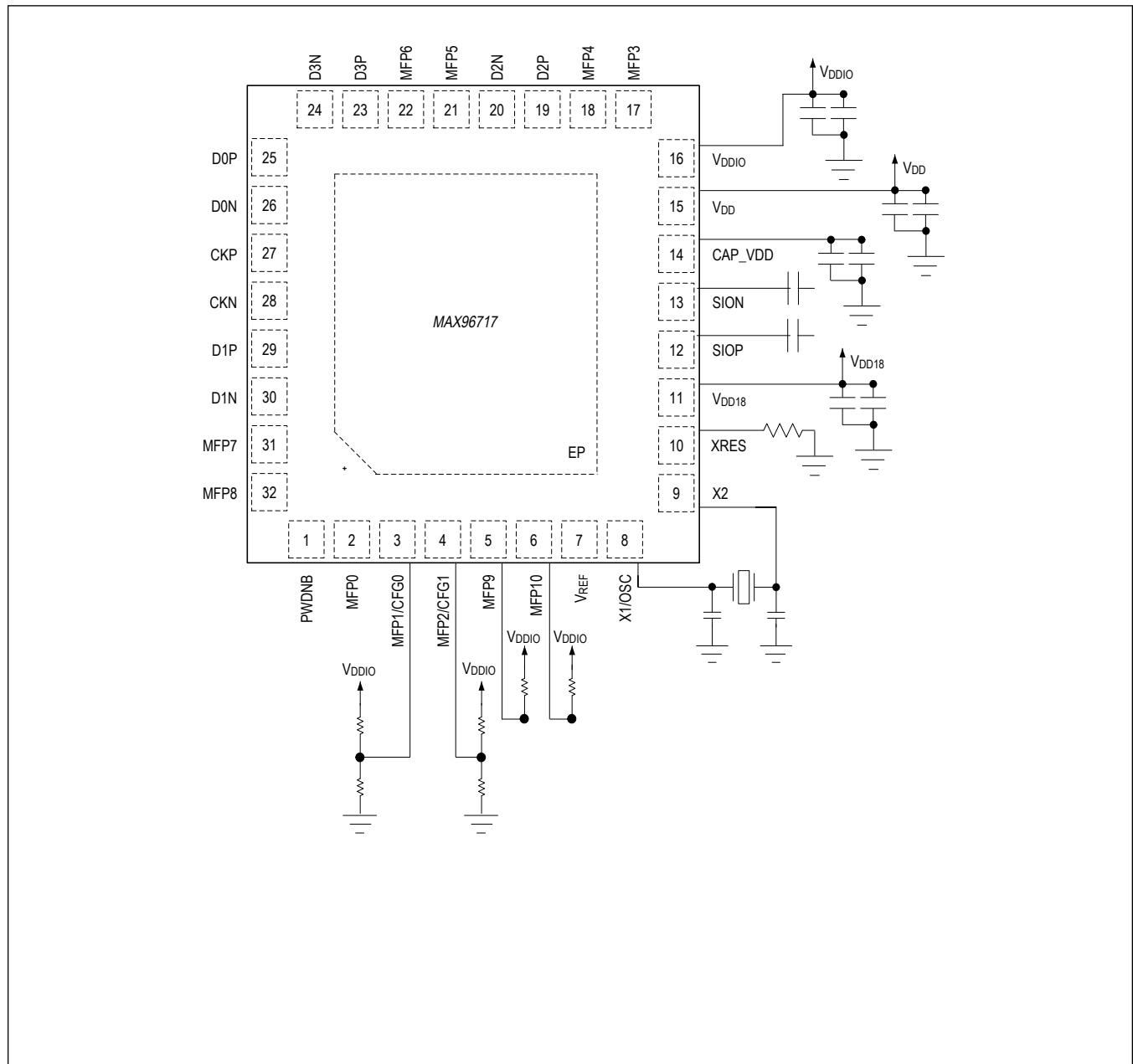


Figure 50. Typical Application Circuit

## Ordering Information

PART NUMBER	TEMP RANGE	PIN-PACKAGE
MAX96717GTJ/VY+	-40°C to +105°C	32 TQFN-SW-EP
MAX96717GTJ/VY+T	-40°C to +105°C	32 TQFN-SW-EP

+ Denotes a lead(Pb)-free/RoHS-compliant package.

T Denotes tape-and-reel.

/V Denotes automotive qualified.

Y Denotes wettable flank.

EP = Exposed pad.

## Register Map

### MAX96717

Not all register bits in the register space are shown in the register table. Any bit not explicitly defined in the register table should be treated as reserved and not modified. When a write is required to a register with both defined and undefined register bits, first read the register's contents, then create a new register value by only changing the defined bits. Finally, write the new byte to the register (Read/Replace/Write).

Default values are provided for read-only register bits. Read-only bit states are changed at power-up according to the actual state of the device. To avoid overwriting these bits, treat read-only bits as undefined.

**Note:** See the Programming Notes section for mandatory register writes on startup.

\*Register is stored in a retention memory when entering sleep mode and is restored upon exit.

ADDRESS	RESET	NAME	MSB							LSB
<b>DEV</b>										
0x00	0x80	<a href="#">REG0[7:0]*</a>								CFG_BL OCK
0x01	0x08	<a href="#">REG1[7:0]*</a>	IIC_2_E N	IIC_1_E N	DIS_LO CAL_CC	DIS_RE M_CC	TX_RATE[1:0]			RX_RATE[1:0]
0x02	0x43	<a href="#">REG2[7:0]*</a>	—	VID_TX_ EN_Z	—	—	—	—	INFO_T X_EN	INFO_R X_EN
0x03	0x00	<a href="#">REG3[7:0]*</a>	—	—	RSVD	UART_1 _EN	—	RCLK_A LT		RCLKSEL[1:0]
0x04	0x18	<a href="#">REG4[7:0]*</a>	—	—	—	CC_MS GCNTR_— EN	CC_CRC _EN	CC_CRC _MSGC NTR_OV R	RSVD	XTAL_P U
0x05	0x00	<a href="#">REG5[7:0]*</a>	LOCK_E N	ERRB_E N	ALT_LO CK_EN	ALT_ER RB_EN	RSVD	RSVD	PU_LF1	PU_LF0
0x06	0x80	<a href="#">REG6[7:0]*</a>	RSVD	—	RCLKEN	I2CSEL	—	—	—	RSVD
0x07	0xF6	<a href="#">REG7[7:0]*</a>				RSVD[3:0]	RSVD	RSVD	RSVD	PAR_VI D_EN
0x0D	0xB7	<a href="#">REG13[7:0]</a>					DEV_ID[7:0]			
0x0E	0x06	<a href="#">REG14[7:0]</a>				RSVD[3:0]			DEV_REV[3:0]	
0x0F	0x00	<a href="#">REG15[7:0]</a>	RSVD	RSVD		SPEED_CPBL[1:0]	RSVD	RSVD	RSVD	RSVD
0x26	0x22	<a href="#">REG26[7:0]</a>	—			LF_1[2:0]	—		LF_0[2:0]	
OVERLAP										
<b>TCTRL</b>										
0x08	0x00	<a href="#">PWR0[7:0]</a>			VDDBAD_STATUS[2:0]				CMP_STATUS[4:0]	
0x09	0x00	<a href="#">PWR1[7:0]</a>	OVERTE MP	VDD_OV _LIVE					PORZ_STATUS[5:0]	
0x0C	0x15	<a href="#">PWR4[7:0]*</a>	RSVD	DIS_LO CAL_WA KE	—	WAKE_E N_A				RSVD[3:0]
0x10	0x01	<a href="#">CTRL0[7:0]*</a>	RESET_ ALL	RESET_ LINK	RESET_ ONESH OT	RSVD	SLEEP	—		RSVD[1:0]
0x11	0x02	<a href="#">CTRL1[7:0]*</a>	RSVD	RSVD	RSVD	—	—	—	RSVD	CXTP_A

ADDRESS	RESET	NAME	MSB						LSB
0x12	0x04	<a href="#">CTRL2[7:0]</a>	RSVD	RSVD	-	LDO_BY_PASS	RSVD[1:0]		RSVD[1:0]
0x13	0x10	<a href="#">CTRL3[7:0]</a>	RSVD	RSVD	RSVD[1:0]		LOCKED	ERROR	CMU_LOCKED
0x18	0xA0	<a href="#">INTR0[7:0]*</a>	RSVD	RSVD	RSVD	-	AUTO_ERR_RST_EN	DEC_ERR_THR[2:0]	
0x19	0x00	<a href="#">INTR1[7:0]*</a>	PKT_CNT_EXP[3:0]			AUTO_CNT_RST_EN	RSVD[2:0]		
0x1A	0x09	<a href="#">INTR2[7:0]*</a>	REFGEN_UNLOC_KED_OE_N	RSVD	REM_ER_R_OEN	-	LFLT_IN_T_OEN	IDLE_ER_R_OEN	-
0x1B	0x00	<a href="#">INTR3[7:0]</a>	REFGEN_UNLOC_KED	RSVD	REM_ER_R_FLAG	-	LFLT_IN_T	IDLE_ER_R_FLAG	-
0x1C	0x08	<a href="#">INTR4[7:0]*</a>	VREG_OV_OEN	EOM_ERR_OEN_A	VDD_OV_OEN	VDD18_OV_OEN	MAX_RT_OEN	RT_CNT_OEN	PKT_CNT_OEN
0x1D	0x00	<a href="#">INTR5[7:0]</a>	VREG_OV_FLAG	EOM_ERR_FLAG_A	VDD_OV_FLAG	VDD18_OV_FLAG	MAX_RT_FLAG	RT_CNT_FLAG	PKT_CNT_FLAG
0x1E	0xFB	<a href="#">INTR6[7:0]*</a>	VDDCM_P_INT_OEN	PORZ_INT_OEN	VDBBAD_INT_OE_N	EFUSE_CRC_ER_R_OEN	RTTN_CRC_ERR_OEN	ADC_IN_T_OEN	MIPI_ER_R_OEN
0x1F	0x00	<a href="#">INTR7[7:0]</a>	VDDCM_P_INT_FLAG	PORZ_INT_FLAG	VDBBAD_INT_FLAG	EFUSE_CRC_ER_R	RTTN_CRC_INT	ADC_IN_T_FLAG	MIPI_ER_R_FLAG
0x20	0x9F	<a href="#">INTR8[7:0]*</a>	ERR_TX_EN	-	-	ERR_TX_ID[4:0]			
0x21	0xDF	<a href="#">INTR9[7:0]*</a>	ERR_RX_EN	RSVD	-	ERR_RX_ID[4:0]			
0x22	0x00	<a href="#">CNT0[7:0]</a>	DEC_ERR_A[7:0]				IDLE_ERR[7:0]		
0x24	0x00	<a href="#">CNT2[7:0]</a>							
0x25	0x00	<a href="#">CNT3[7:0]</a>	PKT_CNT[7:0]						
<b>GMSL</b>									
0x28	0x60	<a href="#">TX0[7:0]*</a>	RSVD[1:0]		RSVD	RSVD	-	-	TX_FEC_EN
0x29	0x08	<a href="#">TX1[7:0]*</a>	LINK_PRBS_GEN	RSVD	-	ERRG_EN_A	TX_FEC_CRC_EN	-	DIS_SC_R
0x2A	0x20	<a href="#">TX2[7:0]*</a>	ERRG_CNT[1:0]		ERRG_RATE[1:0]		ERRG_BURST[2:0]		ERRG_PER
0x2B	0x44	<a href="#">TX3[7:0]*</a>	RSVD[1:0]		TX_FEC_ACTIVE	-	-	RSVD[2:0]	
0x2C	0x00	<a href="#">RX0[7:0]*</a>	PKT_CNT_LBW[1:0]		-	RSVD	PKT_CNT_SEL[3:0]		
0x2D	0x28	<a href="#">RX1[7:0]*</a>	LINK_PRBS_CHK	-	RSVD[1:0]		RSVD[1:0]	RSVD	RSVD
0x30	0x41	<a href="#">GPIOA[7:0]*</a>	RSVD	RSVD	GPIO_FWD_CDLY[5:0]				

ADDRESS	RESET	NAME	MSB							LSB					
0x31	0x88	<a href="#">GPIOB[7:0]*</a>	RSVD[1:0]	GPIO_REV_CDLY[5:0]											
<b>CC</b>															
0x40	0x26	<a href="#">I2C_0[7:0]*</a>	–	–	SLV_SH[1:0]		–	SLV_TO[2:0]							
0x41	0x56	<a href="#">I2C_1[7:0]*</a>	RSVD	MST_BT[2:0]			–	MST_TO[2:0]							
0x42	0x00	<a href="#">I2C_2[7:0]*</a>	SRC_A[6:0]					–							
0x43	0x00	<a href="#">I2C_3[7:0]*</a>	DST_A[6:0]					–							
0x44	0x00	<a href="#">I2C_4[7:0]*</a>	SRC_B[6:0]					–							
0x45	0x00	<a href="#">I2C_5[7:0]*</a>	DST_B[6:0]					–							
0x48	0x42	<a href="#">UART_0[7:0]*</a>	RSVD[1:0]		REM_MS_EN	LOC_MS_EN	BYPASS_DIS_PA_R	BYPASS_TO[1:0]		BYPASS_EN					
0x4C	0x26	<a href="#">I2C_PT_0[7:0]</a> 1	–	–	SLV_SH_PT[1:0]		–	SLV_TO_PT[2:0]							
0x4D	0x56	<a href="#">I2C_PT_1[7:0]</a> 1	RSVD	MST_BT_PT[2:0]			–	MST_TO_PT[2:0]							
0x4F	0x00	<a href="#">UART_PT_0[7:0]</a>	BITLEN_MAN_CF_G_2	DIS_PA_R_2	RSVD	RSVD	BITLEN_MAN_CF_G_1	DIS_PA_R_1	RSVD	RSVD					
<b>CFGV VIDEO_Z</b>															
0x58	0x30	<a href="#">TX0[7:0]*</a>	TX_CRC_EN	–	RSVD[1:0]		RSVD[1:0]		RSVD[1:0]						
0x5B	0x02	<a href="#">TX3[7:0]*</a>	–	–	–	–	–	–	TX_STR_SEL[1:0]						
<b>CFGI INFOFR</b>															
0x78	0xF0	<a href="#">TR0[7:0]*</a>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]		RSVD[1:0]		RSVD[1:0]						
0x7B	0x00	<a href="#">TR3[7:0]*</a>	–	–	–	–	–	TX_SRC_ID[2:0]							
0x7C	0xFF	<a href="#">TR4[7:0]*</a>	RX_SRC_SEL[7:0]												
<b>CFGI SPI</b>															
0x80	0xF0	<a href="#">TR0[7:0]*</a>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]		RSVD[1:0]		RSVD[1:0]						
0x83	0x00	<a href="#">TR3[7:0]*</a>	–	–	–	–	–	TX_SRC_ID[2:0]							
0x84	0xFF	<a href="#">TR4[7:0]*</a>	RX_SRC_SEL[7:0]												
0x85	0x98	<a href="#">ARQ0[7:0]*</a>	RSVD	RSVD	RSVD	RSVD	ARQ0_EN	DIS_DBL_ACK_R_ETX	–	–					
0x86	0x72	<a href="#">ARQ1[7:0]*</a>	–	RSVD[2:0]			–	–	MAX_RT_ERR_O_EN	RT_CNT_OEN					
0x87	0x00	<a href="#">ARQ2[7:0]</a>	MAX_RT_ERR	RT_CNT[6:0]											
<b>CFGC CC</b>															
0x88	0xF0	<a href="#">TR0[7:0]*</a>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]		RSVD[1:0]		RSVD[1:0]						
0x8B	0x00	<a href="#">TR3[7:0]*</a>	–	–	–	–	–	TX_SRC_ID[2:0]							
0x8E	0x72	<a href="#">ARQ1[7:0]*</a>	–	RSVD[2:0]			–	–	MAX_RT_ERR_O_EN	RT_CNT_OEN					

ADDRESS	RESET	NAME	MSB							LSB
0x8F	0x00	<a href="#">ARQ2[7:0]</a>	MAX_RT_ERR	RT_CNT[6:0]						
<b>CFG1 GPIO</b>										
0x90	0xF0	<a href="#">TR0[7:0]*</a>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]		RSVD[1:0]	RSVD[1:0]		
0x93	0x00	<a href="#">TR3[7:0]*</a>	—	—	—	—	—	TX_SRC_ID[2:0]		
0x94	0xFF	<a href="#">TR4[7:0]*</a>	RX_SRC_SEL[7:0]							
0x95	0x98	<a href="#">ARQ0[7:0]*</a>	RSVD	RSVD	RSVD	RSVD	ARQ0_EN	DIS_DBL_ACK_R_ETX	—	—
0x96	0x72	<a href="#">ARQ1[7:0]*</a>	—	RSVD[2:0]			—	—	MAX_RT_ERR_O_EN	RT_CNT_OEN
0x97	0x00	<a href="#">ARQ2[7:0]</a>	MAX_RT_ERR	RT_CNT[6:0]						
<b>CFG1 IIC_X</b>										
0xA0	0xF0	<a href="#">TR0[7:0]*</a>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]		RSVD[1:0]	RSVD[1:0]		
0xA3	0x00	<a href="#">TR3[7:0]*</a>	—	—	—	—	—	TX_SRC_ID[2:0]		
0xA4	0xFF	<a href="#">TR4[7:0]*</a>	RX_SRC_SEL[7:0]							
0xA5	0x98	<a href="#">ARQ0[7:0]*</a>	RSVD	RSVD	RSVD	RSVD	ARQ0_EN	DIS_DBL_ACK_R_ETX	—	—
0xA6	0x72	<a href="#">ARQ1[7:0]*</a>	—	RSVD[2:0]			—	—	MAX_RT_ERR_O_EN	RT_CNT_OEN
0xA7	0x00	<a href="#">ARQ2[7:0]</a>	MAX_RT_ERR	RT_CNT[6:0]						
<b>CFG1 IIC_Y</b>										
0xA8	0xF0	<a href="#">TR0[7:0]*</a>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]		RSVD[1:0]	RSVD[1:0]		
0xAB	0x00	<a href="#">TR3[7:0]*</a>	—	—	—	—	—	TX_SRC_ID[2:0]		
0xAC	0xFF	<a href="#">TR4[7:0]*</a>	RX_SRC_SEL[7:0]							
0xAD	0x98	<a href="#">ARQ0[7:0]*</a>	RSVD	RSVD	RSVD	RSVD	ARQ0_EN	DIS_DBL_ACK_R_ETX	—	—
0xAE	0x72	<a href="#">ARQ1[7:0]*</a>	—	RSVD[2:0]			—	—	MAX_RT_ERR_O_EN	RT_CNT_OEN
0xAF	0x00	<a href="#">ARQ2[7:0]</a>	MAX_RT_ERR	RT_CNT[6:0]						
<b>VID_TX_Z</b>										
0x110	0x68	<a href="#">VIDEO_TX0[7:0]*</a>	LINE_CRC_SEL	LINE_CRC_EN	ENC_MODE[1:0]		AUTO_BPP	CLKDET_BYP	RSVD[1:0]	
0x111	0x58	<a href="#">VIDEO_TX1[7:0]*</a>	RSVD[1:0]		BPP[5:0]					
0x112	0x0A	<a href="#">VIDEO_TX2[7:0]*</a>	PCLKDET	DRIFT_ERR	OVERFL OW	FIFO_WARN	RSVD	LIM_HEART	RSVD	RSVD

ADDRESS	RESET	NAME	MSB							LSB
<b>SPI</b>										
0x170	0x08	<u>SPI_0[7:0]*</u>	SPI_LOC_ID[1:0]	SPI_CC_TRG_ID[1:0]	SPI_IGN_R_ID	SPI_CC_EN	MST_SL_VN	SPI_EN		
0x171	0x1D	<u>SPI_1[7:0]*</u>		SPI_LOC_N[5:0]				SPI_BASE_PRIO[1:0]		
0x172	0x03	<u>SPI_2[7:0]*</u>		REQ_HOLD_OFF[2:0]	FULL_SCK_SET_UP	SPI_MO_D3_F	SPI_MO_D3	SPIM_SS2_ACT_H	SPIM_SS1_ACT_H	
0x173	0x00	<u>SPI_3[7:0]*</u>			SPIM_SS_DLY_CLKS[7:0]					
0x174	0x00	<u>SPI_4[7:0]*</u>			SPIM_SCK_LO_CLKS[7:0]					
0x175	0x00	<u>SPI_5[7:0]*</u>			SPIM_SCK_HI_CLKS[7:0]					
0x176	0x00	<u>SPI_6[7:0]*</u>	–	–	BNE	SPIS_RWN	SS_IO_E_N_2	SS_IO_E_N_1	BNE_IO_EN	RWN_IO_EN
0x177	0x00	<u>SPI_7[7:0]</u>	SPI_RX_OVRFL_W	SPI_TX_OVRFL_W	–		SPIS_BYTECNT[4:0]			
0x178	0x00	<u>SPI_8[7:0]*</u>			REQ_HOLD_OFF_TO[7:0]					
<b>VTX Z</b>										
0x236	0x00	<u>CROSS_0[7:0]*</u>	–	CROSS0_I	CROSS0_F		CROSS0[4:0]			
0x237	0x01	<u>CROSS_1[7:0]*</u>	–	CROSS1_I	CROSS1_F		CROSS1[4:0]			
0x238	0x02	<u>CROSS_2[7:0]*</u>	–	CROSS2_I	CROSS2_F		CROSS2[4:0]			
0x239	0x03	<u>CROSS_3[7:0]*</u>	–	CROSS3_I	CROSS3_F		CROSS3[4:0]			
0x23A	0x04	<u>CROSS_4[7:0]*</u>	–	CROSS4_I	CROSS4_F		CROSS4[4:0]			
0x23B	0x05	<u>CROSS_5[7:0]*</u>	–	CROSS5_I	CROSS5_F		CROSS5[4:0]			
0x23C	0x06	<u>CROSS_6[7:0]*</u>	–	CROSS6_I	CROSS6_F		CROSS6[4:0]			
0x23D	0x07	<u>CROSS_7[7:0]*</u>	–	CROSS7_I	CROSS7_F		CROSS7[4:0]			
0x23E	0x08	<u>CROSS_8[7:0]*</u>	–	CROSS8_I	CROSS8_F		CROSS8[4:0]			
0x23F	0x09	<u>CROSS_9[7:0]*</u>	–	CROSS9_I	CROSS9_F		CROSS9[4:0]			
0x240	0x0A	<u>CROSS_10[7:0]*</u>	–	CROSS10_I	CROSS10_F		CROSS10[4:0]			
0x241	0x0B	<u>CROSS_11[7:0]*</u>	–	CROSS11_I	CROSS11_F		CROSS11[4:0]			
0x242	0x0C	<u>CROSS_12[7:0]*</u>	–	CROSS12_I	CROSS12_F		CROSS12[4:0]			
0x243	0x0D	<u>CROSS_13[7:0]*</u>	–	CROSS13_I	CROSS13_F		CROSS13[4:0]			
0x244	0x0E	<u>CROSS_14[7:0]*</u>	–	CROSS14_I	CROSS14_F		CROSS14[4:0]			

ADDRESS	RESET	NAME	MSB							LSB
0x245	0x0F	<a href="#">CROSS_15[7:0]</a> <sup>*</sup>	—	CROSS1 5_I	CROSS1 5_F					CROSS15[4:0]
0x246	0x10	<a href="#">CROSS_16[7:0]</a> <sup>*</sup>	—	CROSS1 6_I	CROSS1 6_F					CROSS16[4:0]
0x247	0x11	<a href="#">CROSS_17[7:0]</a> <sup>*</sup>	—	CROSS1 7_I	CROSS1 7_F					CROSS17[4:0]
0x248	0x12	<a href="#">CROSS_18[7:0]</a> <sup>*</sup>	—	CROSS1 8_I	CROSS1 8_F					CROSS18[4:0]
0x249	0x13	<a href="#">CROSS_19[7:0]</a> <sup>*</sup>	—	CROSS1 9_I	CROSS1 9_F					CROSS19[4:0]
0x24A	0x14	<a href="#">CROSS_20[7:0]</a> <sup>*</sup>	—	CROSS2 0_I	CROSS2 0_F					CROSS20[4:0]
0x24B	0x15	<a href="#">CROSS_21[7:0]</a> <sup>*</sup>	—	CROSS2 1_I	CROSS2 1_F					CROSS21[4:0]
0x24C	0x16	<a href="#">CROSS_22[7:0]</a> <sup>*</sup>	—	CROSS2 2_I	CROSS2 2_F					CROSS22[4:0]
0x24D	0x17	<a href="#">CROSS_23[7:0]</a> <sup>*</sup>	—	CROSS2 3_I	CROSS2 3_F					CROSS23[4:0]
0x24E	0x03	<a href="#">VTX0[7:0]</a> <sup>*</sup>	GEN_VS	GEN_HS	GEN_DE	VS_INV	HS_INV	DE_INV	VTG_MODE[1:0]	
0x24F	0x01	<a href="#">VTX1[7:0]</a> <sup>*</sup>	—	—	PCLKDET_VTX	—		PATGEN_CLK_SRC[2:0]		VS_TRI_G
0x250	0x00	<a href="#">VTX2[7:0]</a> <sup>*</sup>				VS_DLY_2[7:0]				
0x251	0x00	<a href="#">VTX3[7:0]</a> <sup>*</sup>				VS_DLY_1[7:0]				
0x252	0x00	<a href="#">VTX4[7:0]</a> <sup>*</sup>				VS_DLY_0[7:0]				
0x253	0x00	<a href="#">VTX5[7:0]</a> <sup>*</sup>				VS_HIGH_2[7:0]				
0x254	0x00	<a href="#">VTX6[7:0]</a> <sup>*</sup>				VS_HIGH_1[7:0]				
0x255	0x00	<a href="#">VTX7[7:0]</a> <sup>*</sup>				VS_HIGH_0[7:0]				
0x256	0x00	<a href="#">VTX8[7:0]</a> <sup>*</sup>				VS_LOW_2[7:0]				
0x257	0x00	<a href="#">VTX9[7:0]</a> <sup>*</sup>				VS_LOW_1[7:0]				
0x258	0x00	<a href="#">VTX10[7:0]</a> <sup>*</sup>				VS_LOW_0[7:0]				
0x259	0x00	<a href="#">VTX11[7:0]</a> <sup>*</sup>				V2H_2[7:0]				
0x25A	0x00	<a href="#">VTX12[7:0]</a> <sup>*</sup>				V2H_1[7:0]				
0x25B	0x00	<a href="#">VTX13[7:0]</a> <sup>*</sup>				V2H_0[7:0]				
0x25C	0x00	<a href="#">VTX14[7:0]</a> <sup>*</sup>				HS_HIGH_1[7:0]				
0x25D	0x00	<a href="#">VTX15[7:0]</a> <sup>*</sup>				HS_HIGH_0[7:0]				
0x25E	0x00	<a href="#">VTX16[7:0]</a> <sup>*</sup>				HS_LOW_1[7:0]				
0x25F	0x00	<a href="#">VTX17[7:0]</a> <sup>*</sup>				HS_LOW_0[7:0]				
0x260	0x00	<a href="#">VTX18[7:0]</a> <sup>*</sup>				HS_CNT_1[7:0]				
0x261	0x00	<a href="#">VTX19[7:0]</a> <sup>*</sup>				HS_CNT_0[7:0]				
0x262	0x00	<a href="#">VTX20[7:0]</a> <sup>*</sup>				V2D_2[7:0]				
0x263	0x00	<a href="#">VTX21[7:0]</a> <sup>*</sup>				V2D_1[7:0]				
0x264	0x00	<a href="#">VTX22[7:0]</a> <sup>*</sup>				V2D_0[7:0]				
0x265	0x00	<a href="#">VTX23[7:0]</a> <sup>*</sup>				DE_HIGH_1[7:0]				
0x266	0x00	<a href="#">VTX24[7:0]</a> <sup>*</sup>				DE_HIGH_0[7:0]				
0x267	0x00	<a href="#">VTX25[7:0]</a> <sup>*</sup>				DE_LOW_1[7:0]				

ADDRESS	RESET	NAME	MSB							LSB
0x268	0x00	<a href="#">VTX26[7:0]*</a>								DE_LOW_0[7:0]
0x269	0x00	<a href="#">VTX27[7:0]*</a>								DE_CNT_1[7:0]
0x26A	0x00	<a href="#">VTX28[7:0]*</a>								DE_CNT_0[7:0]
0x26B	0x00	<a href="#">VTX29[7:0]</a>	VID_PR BS_EN	RSVD	VPRBS_ FAIL	-	-	GRAD_ MODE	PATGEN_MODE[1:0]	
0x26C	0x04	<a href="#">VTX30[7:0]</a>						GRAD_INC[7:0]		
0x26D	0x00	<a href="#">VTX31[7:0]</a>						CHKR_A_L[7:0]		
0x26E	0x00	<a href="#">VTX32[7:0]</a>						CHKR_A_M[7:0]		
0x26F	0x00	<a href="#">VTX33[7:0]</a>						CHKR_A_H[7:0]		
0x270	0x00	<a href="#">VTX34[7:0]</a>						CHKR_B_L[7:0]		
0x271	0x00	<a href="#">VTX35[7:0]</a>						CHKR_B_M[7:0]		
0x272	0x00	<a href="#">VTX36[7:0]</a>						CHKR_B_H[7:0]		
0x273	0x00	<a href="#">VTX37[7:0]</a>						CHKR_RPT_A[7:0]		
0x274	0x00	<a href="#">VTX38[7:0]</a>						CHKR_RPT_B[7:0]		
0x275	0x00	<a href="#">VTX39[7:0]</a>						CHKR_ALT[7:0]		
0x276	0x18	<a href="#">VTX40[7:0]</a>	RSVD	CROSS HS_I	CROSS HS_F			CROSSHS[4:0]		
0x277	0x19	<a href="#">VTX41[7:0]</a>	-	CROSS VS_I	CROSS VS_F			CROSSVS[4:0]		
0x278	0x1A	<a href="#">VTX42[7:0]</a>	-	CROSS DE_I	CROSS DE_F			CROSSDE[4:0]		
<b>GPIO0 0</b>										
0x2BE	0x99	<a href="#">GPIO_A[7:0]*</a>	RES_CF G	RSVD	TX_COM P_EN	GPIO_O UT	GPIO_IN	GPIO_R X_EN	GPIO_T X_EN	GPIO_O UT_DIS
0x2BF	0xA0	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1 :0]		OUT_TY PE			GPIO_TX_ID[4:0]		
0x2C0	0x40	<a href="#">GPIO_C[7:0]*</a>	OVR_RE S_CFG	GPIO_R ECVED	-			GPIO_RX_ID[4:0]		
<b>GPIO1 1</b>										
0x2C1	0x81	<a href="#">GPIO_A[7:0]*</a>	RES_CF G	RSVD	TX_COM P_EN	GPIO_O UT	GPIO_IN	GPIO_R X_EN	GPIO_T X_EN	GPIO_O UT_DIS
0x2C2	0x21	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1 :0]		OUT_TY PE			GPIO_TX_ID[4:0]		
0x2C3	0x41	<a href="#">GPIO_C[7:0]*</a>	OVR_RE S_CFG	GPIO_R ECVED	-			GPIO_RX_ID[4:0]		
<b>GPIO2 2</b>										
0x2C4	0x99	<a href="#">GPIO_A[7:0]*</a>	RES_CF G	RSVD	TX_COM P_EN	GPIO_O UT	GPIO_IN	GPIO_R X_EN	GPIO_T X_EN	GPIO_O UT_DIS
0x2C5	0x22	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1 :0]		OUT_TY PE			GPIO_TX_ID[4:0]		
0x2C6	0x42	<a href="#">GPIO_C[7:0]*</a>	OVR_RE S_CFG	GPIO_R ECVED	-			GPIO_RX_ID[4:0]		
<b>GPIO3 3</b>										
0x2C7	0x81	<a href="#">GPIO_A[7:0]*</a>	RES_CF G	RSVD	TX_COM P_EN	GPIO_O UT	GPIO_IN	GPIO_R X_EN	GPIO_T X_EN	GPIO_O UT_DIS
0x2C8	0xA3	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1 :0]		OUT_TY PE			GPIO_TX_ID[4:0]		

ADDRESS	RESET	NAME	MSB							LSB	
0x2C9	0x43	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO4 4</b>											
0x2CA	0x99	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	
0x2CB	0xA4	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY_PĒ	GPIO_TX_ID[4:0]						
0x2CC	0x44	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO5 5</b>											
0x2CD	0x81	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	
0x2CE	0xA5	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY_PĒ	GPIO_TX_ID[4:0]						
0x2CF	0x45	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO6 6</b>											
0x2D0	0x99	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	
0x2D1	0xA6	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY_PĒ	GPIO_TX_ID[4:0]						
0x2D2	0x46	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO7 7</b>											
0x2D3	0x83	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	
0x2D4	0xA7	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY_PĒ	GPIO_TX_ID[4:0]						
0x2D5	0x47	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO8 8</b>											
0x2D6	0x9C	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	
0x2D7	0x28	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY_PĒ	GPIO_TX_ID[4:0]						
0x2D8	0x48	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO9 9</b>											
0x2D9	0x81	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	
0x2DA	0xA9	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY_PĒ	GPIO_TX_ID[4:0]						
0x2DB	0x49	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R_ECVED	–	GPIO_RX_ID[4:0]					
<b>GPIO10 10</b>											
0x2DC	0x99	<a href="#">GPIO_A[7:0]*</a>	RES_CF_G	RSVD	TX_COM_P_EN	GPIO_O_UT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_O_UT_DIS	

ADDRESS	RESET	NAME	MSB							LSB	
0x2DD	0x2A	<a href="#">GPIO_B[7:0]*</a>	PULL_UPDN_SEL[1:0]	OUT_TY PE	GPIO_TX_ID[4:0]						
0x2DE	0x4A	<a href="#">GPIO_C[7:0]*</a>	OVR_RS_CFG	GPIO_R ECVED	—	GPIO_RX_ID[4:0]					
<b>CMU</b>											
0x302	0x00	<a href="#">CMU2[7:0]</a>	RSVD	PFDDIV_RSHORT[2:0]			RSVD	RSVD[1:0]		RSVD	
<b>FRONTTOP</b>											
0x308	0x64	<a href="#">FRONTTOP_0[7:0]*</a>	RSVD	enable_line_info	START_PORTB	—	—	RSVD	—	—	
0x30D	0xFF	<a href="#">FRONTTOP_5[7:0]*</a>	VC_SELZ_L[7:0]								
0x30E	0xFF	<a href="#">FRONTTOP_6[7:0]*</a>	VC_SELZ_H[7:0]								
0x311	0x40	<a href="#">FRONTTOP_9[7:0]*</a>	—	START_PORTBZ	—	—	—	—	—	—	
0x312	0x00	<a href="#">FRONTTOP_10[7:0]*</a>	—	RSVD	—	—	—	bpp8dblz	—	—	
0x313	0x00	<a href="#">FRONTTOP_11[7:0]*</a>	—	bpp12dblz	—	—	—	bpp10dblz	—	—	
0x318	0x00	<a href="#">FRONTTOP_16[7:0]</a>	—	mem_dt1_selz[6:0]							
0x319	0x00	<a href="#">FRONTTOP_17[7:0]</a>	—	mem_dt2_selz[6:0]							
0x31E	0x18	<a href="#">FRONTTOP_22[7:0]</a>	soft_dtz_en	soft_vczi_en	soft_bppz_en	soft_bppz[4:0]					
0x320	0x00	<a href="#">FRONTTOP_24[7:0]</a>	—	—	soft_vczi[1:0]	—	—	—	—	—	
0x323	0x30	<a href="#">FRONTTOP_27[7:0]</a>	—	—	soft_dtz[5:0]						
0x325	0x00	<a href="#">FRONTTOP_29[7:0]</a>	FORCE_START_MIPI_FR ONTTOP	—	—	—	—	—	—	—	
<b>MIPI_RX</b>											
0x330	0x00	<a href="#">MIPI_RX0[7:0]</a>	—	mipi_non contclk_en	ctrl1_vc_map_en	—	mipi_rx_r eset	RSVD[2:0]			
0x331	0x30	<a href="#">MIPI_RX1[7:0]</a>	ctrl1_vcxi_en	ctrl1_des kewen	ctrl1_num_lanes[1:0]		—	—	—	—	
0x332	0xE0	<a href="#">MIPI_RX2[7:0]</a>	phy1_lane_map[3:0]				—	—	—	—	
0x333	0x04	<a href="#">MIPI_RX3[7:0]</a>	—	—	—	—	phy2_lane_map[3:0]				
0x334	0x00	<a href="#">MIPI_RX4[7:0]</a>	—	phy1_pol_map[2:0]			—	—	—	—	
0x335	0x00	<a href="#">MIPI_RX5[7:0]</a>	—	—	—	—	phy2_pol_map[2:0]				
0x337	0x00	<a href="#">MIPI_RX7[7:0]</a>	—	—	ctrl1_mo de_UTD	RSVD[4:0]					

ADDRESS	RESET	NAME	MSB						LSB		
0x338	0x55	<a href="#">MIPI_RX8[7:0]</a>	RSVD[1:0]		t_hs_settle[1:0]		RSVD[1:0]		t_clk_settle[1:0]		
0x33B	0x00	<a href="#">MIPI_RX11[7:0]</a>	-	-	-	phy1_lp_err[4:0]					
0x33C	0x00	<a href="#">MIPI_RX12[7:0]</a>	phy1_hs_err[7:0]								
0x33D	0x00	<a href="#">MIPI_RX13[7:0]</a>	-	-	phy2_lp_err[4:0]						
0x33E	0x00	<a href="#">MIPI_RX14[7:0]</a>	phy2_hs_err[7:0]								
0x343	0x00	<a href="#">MIPI_RX19[7:0]</a>				ctrl1_csi_err_l[7:0]					
0x344	0x00	<a href="#">MIPI_RX20[7:0]</a>	-	-	-	-	-	ctrl1_csi_err_h[2:0]			
0x345	0x00	<a href="#">MIPI_RX21[7:0]</a>	ctrl1_vc_map0[3:0]			-	-	-	-		
0x346	0x00	<a href="#">MIPI_RX22[7:0]</a>	ctrl1_vc_map1[3:0]			-	-	-	-		
0x347	0x00	<a href="#">MIPI_RX23[7:0]</a>	ctrl1_vc_map2[3:0]			-	-	-	-		
0x36C	0x00	<a href="#">MIPI_RX60[7:0]</a>	ctrl1_vc_map3[3:0]			-	-	-	-		
0x36D	0x00	<a href="#">MIPI_RX61[7:0]</a>	ctrl1_vc_map4[3:0]			-	-	-	-		
0x36E	0x00	<a href="#">MIPI_RX62[7:0]</a>	ctrl1_vc_map5[3:0]			-	-	-	-		
0x36F	0x00	<a href="#">MIPI_RX63[7:0]</a>	ctrl1_vc_map6[3:0]			-	-	-	-		
<b>MIPI_RX_EXT</b>											
0x377	0x00	<a href="#">EXT00[7:0]</a>	ctrl1_vc_map7[3:0]			-	-	-	-		
0x378	0x00	<a href="#">EXT0[7:0]</a>	ctrl1_vc_map8[3:0]			-	-	-	-		
0x379	0x00	<a href="#">EXT1[7:0]</a>	ctrl1_vc_map9[3:0]			-	-	-	-		
0x37A	0x00	<a href="#">EXT2[7:0]</a>	ctrl1_vc_map10[3:0]			-	-	-	-		
0x37B	0x00	<a href="#">EXT3[7:0]</a>	ctrl1_vc_map11[3:0]			-	-	-	-		
0x37C	0x00	<a href="#">EXT4[7:0]</a>	ctrl1_vc_map12[3:0]			-	-	-	-		
0x37D	0x00	<a href="#">EXT5[7:0]</a>	ctrl1_vc_map13[3:0]			-	-	-	-		
0x37E	0x00	<a href="#">EXT6[7:0]</a>	ctrl1_vc_map14[3:0]			-	-	-	-		
0x37F	0x00	<a href="#">EXT7[7:0]</a>	ctrl1_vc_map15[3:0]			-	-	-	-		
0x380	0x00	<a href="#">EXT8[7:0]</a>	RSVD	RSVD[1:0]		RSVD[1:0]		RSVD	RSVD		
0x381	0x00	<a href="#">EXT9[7:0]</a>	RSVD[7:0]								
0x383	0x80	<a href="#">EXT11[7:0]</a>	Tun_Mode	RSVD	-	-	RSVD	RSVD	RSVD[1:0]		
0x38D	0x00	<a href="#">EXT21[7:0]</a>	phy1_pkt_cnt[7:0]								
0x38E	0x00	<a href="#">EXT22[7:0]</a>	csi1_pkt_cnt[7:0]								
0x38F	0x00	<a href="#">EXT23[7:0]</a>	tun_pkt_cnt[7:0]								
0x390	0x00	<a href="#">EXT24[7:0]</a>	phy_clk_cnt[7:0]								

ADDRESS	RESET	NAME	MSB							LSB
<b>FRONTTOP_EXT</b>										
0x3C8	0x00	<a href="#">FRONTTOP_EXT8[7:0]</a>								mem_dt3_selz[7:0]
0x3C9	0x00	<a href="#">FRONTTOP_EXT9[7:0]</a>								mem_dt4_selz[7:0]
0x3CA	0x00	<a href="#">FRONTTOP_EXT10[7:0]</a>								mem_dt5_selz[7:0]
0x3CB	0x00	<a href="#">FRONTTOP_EXT11[7:0]</a>								mem_dt6_selz[7:0]
0x3D1	0x00	<a href="#">FRONTTOP_EXT17[7:0]</a>	-	-	-	-	mem_dt6_selz_en	mem_dt5_selz_en	mem_dt4_selz_en	mem_dt3_selz_en
<b>MIPI_RX_EXT2</b>										
0x3DC	0x00	<a href="#">EXTA[7:0]</a>	-							mem_dt7_selz[6:0]
0x3DD	0x00	<a href="#">EXTB[7:0]</a>	-							mem_dt8_selz[6:0]
<b>REF_VTG</b>										
0x3E0	0x70	<a href="#">VTX0[7:0]</a>	-	VS_TRI_G	REF_VTG_MODE[1:0]	HS_INV	GEN_HS	VS_INV	GEN_VS	
0x3E1	0x00	<a href="#">VTX1[7:0]</a>								VS_HIGH_2[7:0]
0x3E2	0x00	<a href="#">VTX2[7:0]</a>								VS_HIGH_1[7:0]
0x3E3	0x00	<a href="#">VTX3[7:0]</a>								VS_HIGH_0[7:0]
0x3E4	0x00	<a href="#">VTX4[7:0]</a>								VS_LOW_2[7:0]
0x3E5	0x00	<a href="#">VTX5[7:0]</a>								VS_LOW_1[7:0]
0x3E6	0x00	<a href="#">VTX6[7:0]</a>								VS_LOW_0[7:0]
0x3E7	0x00	<a href="#">VTX7[7:0]</a>								V2H_2[7:0]
0x3E8	0x00	<a href="#">VTX8[7:0]</a>								V2H_1[7:0]
0x3E9	0x00	<a href="#">VTX9[7:0]</a>								V2H_0[7:0]
0x3EA	0x00	<a href="#">VTX10[7:0]</a>								HS_HIGH_1[7:0]
0x3EB	0x00	<a href="#">VTX11[7:0]</a>								HS_HIGH_0[7:0]
0x3EC	0x00	<a href="#">VTX12[7:0]</a>								HS_LOW_1[7:0]
0x3ED	0x00	<a href="#">VTX13[7:0]</a>								HS_LOW_0[7:0]
0x3EE	0x00	<a href="#">VTX14[7:0]</a>								HS_CNT_1[7:0]
0x3EF	0x00	<a href="#">VTX15[7:0]</a>								HS_CNT_0[7:0]
0x3F0	0x50	<a href="#">REF_VTG0[7:0]</a>	REFGEN_LOCKED	REFGEN_PREDEF_EN	REFGEN_PREDEF_FREQ[1:0]	REFGEN_PREDEF_FREQ_ALT	-	REFGEN_RST	REFGEN_EN	
0x3F4	0x00	<a href="#">REF_VTG4[7:0]</a>								REFGEN_FB_FRACT_L[7:0]
0x3F5	0x00	<a href="#">REF_VTG5[7:0]</a>	-	-	-	-				REFGEN_FB_FRACT_H[3:0]
0x3F6	0x00	<a href="#">REF_VTG6[7:0]</a>								VS_DLY_2[7:0]
0x3F7	0x00	<a href="#">REF_VTG7[7:0]</a>								VS_DLY_1[7:0]
0x3F8	0x00	<a href="#">REF_VTG8[7:0]</a>								VS_DLY_0[7:0]

ADDRESS	RESET	NAME	MSB							LSB		
0x3F9	0x1E	<a href="#">REF_VTG9[7:0]</a>	REF_VT G_TRIG _EN	–	–	REF_VTG_TRIG_ID[4:0]						
<b>AFE</b>												
0x500	0x00	<a href="#">ADC_CTRL_0[7:0]</a>	buf_bypass	RSVD	RSVD	adc_chg_pump_pu	adc_refbuf_pu	buf_pu	adc_pu	cpu_adc_start		
0x501	0x00	<a href="#">ADC_CTRL_1[7:0]</a>	adc_chsel[3:0]				adc_clk_en	adc_refsel	adc_scale	RSVD		
0x502	0x00	<a href="#">ADC_CTRL_2[7:0]</a>	RSVD	–	RSVD	RSVD	adc_div[1:0]		adc_xref	Inmux_en		
0x508	0x00	<a href="#">ADC_DATA0[7:0]</a>	adc_data_l[7:0]									
0x509	0x00	<a href="#">ADC_DATA1[7:0]</a>	bypass_volttemp_corr	–	–	–	–	–	adc_data_h[1:0]			
0x50C	0x00	<a href="#">ADC_INTRIE0[7:0]</a>	adc_caldone_ie	adc_overRange_ie	adc_tmon_cal_ood_ie	RSVD	adc_lo_limit_ie	adc_hi_limit_ie	adc_ref_ready_ie	adc_donne_ie		
0x50D	0x00	<a href="#">ADC_INTRIE1[7:0]</a>	ch7_hi_limit_ie	ch6_hi_limit_ie	ch5_hi_limit_ie	ch4_hi_limit_ie	ch3_hi_limit_ie	ch2_hi_limit_ie	ch1_hi_limit_ie	ch0_hi_limit_ie		
0x50E	0x00	<a href="#">ADC_INTRIE2[7:0]</a>	ch7_lo_limit_ie	ch6_lo_limit_ie	ch5_lo_limit_ie	ch4_lo_limit_ie	ch3_lo_limit_ie	ch2_lo_limit_ie	ch1_lo_limit_ie	ch0_lo_limit_ie		
0x50F	0x00	<a href="#">ADC_INTRIE3[7:0]</a>	–	reflim_ie	reflimscl1_ie	reflimscl2_ie	reflimscl3_ie	RSVD	tmon_err_ie	RSVD		
0x510	0x00	<a href="#">ADC_INTR0[7:0]</a>	adc_caldone_if	adc_overRange_if	adc_tmon_cal_ood_if	RSVD	adc_lo_limit_if	adc_hi_limit_if	adc_ref_ready_if	adc_donne_if		
0x511	0x00	<a href="#">ADC_INTR1[7:0]</a>	ch7_hi_limit_if	ch6_hi_limit_if	ch5_hi_limit_if	ch4_hi_limit_if	ch3_hi_limit_if	ch2_hi_limit_if	ch1_hi_limit_if	ch0_hi_limit_if		
0x512	0x00	<a href="#">ADC_INTR2[7:0]</a>	ch7_lo_limit_if	ch6_lo_limit_if	ch5_lo_limit_if	ch4_lo_limit_if	ch3_lo_limit_if	ch2_lo_limit_if	ch1_lo_limit_if	ch0_lo_limit_if		
0x513	0x00	<a href="#">ADC_INTR3[7:0]</a>	–	reflim_if	reflimscl1_if	reflimscl2_if	reflimscl3_if	RSVD	tmon_err_if	RSVD		
0x514	0x00	<a href="#">ADC_LIMIT0[7:0]</a>	chLoLimit_l0[7:0]									
0x515	0xF0	<a href="#">ADC_LIMIT0_1[7:0]</a>	chHiLimit_l0[3:0]				–	–	chLoLimit_h0[1:0]			
0x516	0x3F	<a href="#">ADC_LIMIT0_2[7:0]</a>	–	–	chHiLimit_h0[5:0]							
0x517	0x03	<a href="#">ADC_LIMIT0_3[7:0]</a>	–	–	div_sel0[1:0]		ch_sel0[3:0]					
0x518	0x00	<a href="#">ADC_LIMIT1_0[7:0]</a>	chLoLimit_l1[7:0]									
0x519	0xF0	<a href="#">ADC_LIMIT1_1[7:0]</a>	chHiLimit_l1[3:0]				–	–	chLoLimit_h1[1:0]			
0x51A	0x3F	<a href="#">ADC_LIMIT1_2[7:0]</a>	–	–	chHiLimit_h1[5:0]							
0x51B	0x03	<a href="#">ADC_LIMIT1_3[7:0]</a>	–	–	div_sel1[1:0]		ch_sel1[3:0]					

ADDRESS	RESET	NAME	MSB							LSB
0x51C	0x00	<u>ADC_LIMIT2_0[7:0]</u>								chLoLimit_I2[7:0]
0x51D	0xF0	<u>ADC_LIMIT2_1[7:0]</u>			chHiLimit_I2[3:0]		–	–		chLoLimit_h2[1:0]
0x51E	0x3F	<u>ADC_LIMIT2_2[7:0]</u>	–	–						chHiLimit_h2[5:0]
0x51F	0x03	<u>ADC_LIMIT2_3[7:0]</u>	–	–	div_sel2[1:0]					ch_sel2[3:0]
0x520	0x00	<u>ADC_LIMIT3_0[7:0]</u>					chLoLimit_I3[7:0]			
0x521	0xF0	<u>ADC_LIMIT3_1[7:0]</u>			chHiLimit_I3[3:0]		–	–		chLoLimit_h3[1:0]
0x522	0x3F	<u>ADC_LIMIT3_2[7:0]</u>	–	–						chHiLimit_h3[5:0]
0x523	0x03	<u>ADC_LIMIT3_3[7:0]</u>	–	–	div_sel3[1:0]					ch_sel3[3:0]
0x524	0x00	<u>ADC_LIMIT4_0[7:0]</u>					chLoLimit_I4[7:0]			
0x525	0xF0	<u>ADC_LIMIT4_1[7:0]</u>			chHiLimit_I4[3:0]		–	–		chLoLimit_h4[1:0]
0x526	0x3F	<u>ADC_LIMIT4_2[7:0]</u>	–	–						chHiLimit_h4[5:0]
0x527	0x03	<u>ADC_LIMIT4_3[7:0]</u>	–	–	div_sel4[1:0]					ch_sel4[3:0]
0x528	0x00	<u>ADC_LIMIT5_0[7:0]</u>					chLoLimit_I5[7:0]			
0x529	0xF0	<u>ADC_LIMIT5_1[7:0]</u>			chHiLimit_I5[3:0]		–	–		chLoLimit_h5[1:0]
0x52A	0x3F	<u>ADC_LIMIT5_2[7:0]</u>	–	–						chHiLimit_h5[5:0]
0x52B	0x03	<u>ADC_LIMIT5_3[7:0]</u>	–	–	div_sel5[1:0]					ch_sel5[3:0]
0x52C	0x00	<u>ADC_LIMIT6_0[7:0]</u>					chLoLimit_I6[7:0]			
0x52D	0xF0	<u>ADC_LIMIT6_1[7:0]</u>			chHiLimit_I6[3:0]		–	–		chLoLimit_h6[1:0]
0x52E	0x3F	<u>ADC_LIMIT6_2[7:0]</u>	–	–						chHiLimit_h6[5:0]
0x52F	0x03	<u>ADC_LIMIT6_3[7:0]</u>	–	–	div_sel6[1:0]					ch_sel6[3:0]
0x530	0x00	<u>ADC_LIMIT7_0[7:0]</u>					chLoLimit_I7[7:0]			
0x531	0xF0	<u>ADC_LIMIT7_1[7:0]</u>			chHiLimit_I7[3:0]		–	–		chLoLimit_h7[1:0]
0x532	0x3F	<u>ADC_LIMIT7_2[7:0]</u>	–	–						chHiLimit_h7[5:0]
0x533	0x03	<u>ADC_LIMIT7_3[7:0]</u>	–	–	div_sel7[1:0]					ch_sel7[3:0]
0x534	0x00	<u>ADC_RR_CT_RL0[7:0]</u>	–	–	–	–	–	–	–	adc_rr_r un

ADDRESS	RESET	NAME	MSB							LSB
0x536	0x00	<a href="#">ADC_RR_CT</a> RL2[7:0]								adc_rr_sleep_l[7:0]
0x537	0x00	<a href="#">ADC_RR_CT</a> RL3[7:0]								adc_rr_sleep_h[7:0]
0x53E	0x00	<a href="#">ADC_CTRL_4</a> [7:0]	RSVD		RSVD[3:0]					adc_pin_en[2:0]
<b>MISC</b>										
0x548	0xDC	<a href="#">UART_PT_0</a> [7:0]*								BITLEN_PT_1_L[7:0]
0x549	0x05	<a href="#">UART_PT_1</a> [7:0]*	-	-						BITLEN_PT_1_H[5:0]
0x54A	0xDC	<a href="#">UART_PT_2</a> [7:0]*								BITLEN_PT_2_L[7:0]
0x54B	0x05	<a href="#">UART_PT_3</a> [7:0]*	-	-						BITLEN_PT_2_H[5:0]
0x550	0x00	<a href="#">I2C_PT_4</a> [7:0] 1								SRC_A_1[6:0]
0x551	0x00	<a href="#">I2C_PT_5</a> [7:0] 1								DST_A_1[6:0]
0x552	0x00	<a href="#">I2C_PT_6</a> [7:0] 1								SRC_B_1[6:0]
0x553	0x00	<a href="#">I2C_PT_7</a> [7:0] 1								DST_B_1[6:0]
0x554	0x00	<a href="#">I2C_PT_8</a> [7:0] 1								SRC_A_2[6:0]
0x555	0x00	<a href="#">I2C_PT_9</a> [7:0] 1								DST_A_2[6:0]
0x556	0x00	<a href="#">I2C_PT_10</a> [7: 0]								SRC_B_2[6:0]
0x557	0x00	<a href="#">I2C_PT_11</a> [7: 0]								DST_B_2[6:0]
0x55F	0x00	<a href="#">HS_VS_Z</a> [7:0]	-	DE_DET_Z	VS_DET_Z	HS_DET_Z	-	-	VS_POL_Z	HS_POL_Z
0x56E	0xBB	<a href="#">UNLOCK_KEY</a> [7:0]*								UNLOCK_KEY[7:0]
0x56F	0x3E	<a href="#">PIO_SLEW_0</a> [7:0]*	-	-	PIO02_SLEW[1:0]	PIO01_SLEW[1:0]	PIO00_SLEW[1:0]			
0x570	0x3C	<a href="#">PIO_SLEW_1</a> [7:0]*	-	-	PIO06_SLEW[1:0]	PIO05_SLEW[1:0]	-	-		
0x571	0xFC	<a href="#">PIO_SLEW_2</a> [7:0]*	PIO011_SLEW[1:0]	PIO010_SLEW[1:0]	RSVD[1:0]	-	-			
<b>MIPI_RX_EXT3</b>										
0x584	0x00	<a href="#">EXT4</a> [7:0]								ctrl1_fs_cnt_l[7:0]
0x585	0x00	<a href="#">EXT5</a> [7:0]								ctrl1_fs_cnt_h[7:0]
0x586	0x00	<a href="#">EXT6</a> [7:0]								ctrl1_fe_cnt_l[7:0]
0x587	0x00	<a href="#">EXT7</a> [7:0]								ctrl1_fe_cnt_h[7:0]
0x588	0x00	<a href="#">EXT8</a> [7:0]	-	-	-	-				ctrl1_fs_vc_sel[3:0]

ADDRESS	RESET	NAME	MSB							LSB		
<b>SPI_CC_WR</b>												
0x1300	0x00	<a href="#">SPI_CC_WR[7:0]</a>	-	-	-	-	-	-	-	-		
<b>SPI_CC_RD</b>												
0x1380	0x00	<a href="#">SPI_CC_RD[7:0]</a>	-	-	-	-	-	-	-	-		
<b>RLMS A</b>												
0x1403	0x0A	<a href="#">RLMS3[7:0]</a>	AdaptEn	AdpHaltL OS	AdpStart Trn	AdpTimH Id	GearShE n	-	GearSh[1:0]			
0x1404	0x4B	<a href="#">RLMS4[7:0]*</a>	EOM_CHK_AMOUNT[3:0]				EOM_CHK_THR[1:0]		EOM_PE R_MOD E	EOM_E N		
0x1405	0x10	<a href="#">RLMS5[7:0]*</a>	EOM_M AN_TRG _REQ	EOM_MIN_THR[6:0]								
0x1406	0x80	<a href="#">RLMS6[7:0]*</a>	EOM_PV _MODE	RSVD[6:0]								
0x1407	0x00	<a href="#">RLMS7[7:0]</a>	EOM_D ONE	EOM[6:0]								
0x1417	0x00	<a href="#">RLMS17[7:0]</a>	DFE5En	DFE4En	DFE3En	DFE2En	DFE1En	BSTEnO v	BSTEn	AGCEn		
0x141C	0x00	<a href="#">RLMS1C[7:0]</a>	AGCMuL[7:0]									
0x141D	0x02	<a href="#">RLMS1D[7:0]</a>	-	-	AGCMuH[5:0]							
0x141F	0x00	<a href="#">RLMS1F[7:0]</a>	AGCInit[7:0]									
0x1432	0x7F	<a href="#">RLMS32[7:0]</a>	OSNMod e	RSVD	RSVD[5:0]							
0x143A	0x00	<a href="#">RLMS3A[7:0]</a>	EyeMonValCntL[7:0]									
0x143B	0x00	<a href="#">RLMS3B[7:0]</a>	EyeMonValCntH[7:0]									
0x1464	0x90	<a href="#">RLMS64[7:0]*</a>	RSVD[3:0]				-	RSVD	TxSSCMode[1:0]			
0x1470	0x01	<a href="#">RLMS70[7:0]*</a>	-	TxSSCFrqCtrl[6:0]								
0x1471	0x02	<a href="#">RLMS71[7:0]*</a>	-	TxSSCCenSprSt[5:0]							TxSSCE n	
0x1472	0xCF	<a href="#">RLMS72[7:0]*</a>	TxSSCPreSclL[7:0]									
0x1473	0x00	<a href="#">RLMS73[7:0]*</a>	-	-	-	-	-	-	TxSSCPreSclH[2:0]			
0x1474	0x00	<a href="#">RLMS74[7:0]*</a>	TxSSCPhL[7:0]									
0x1475	0x00	<a href="#">RLMS75[7:0]*</a>	-	TxSSCPhH[6:0]								
0x1476	0x00	<a href="#">RLMS76[7:0]*</a>	-	-	-	-	-	-	TxSSCPhQuad[1:0]			
0x1484	0x00	<a href="#">RLMS84[7:0]</a>	TuneAm p_ManL	TuneOff_Man[6:0]								
0x1485	0x3B	<a href="#">RLMS85[7:0]</a>	TuneAmp_ManH[7:0]									
0x1495	0x69	<a href="#">RLMS95[7:0]</a>	TxAmpl ManEn	ADCSar Method	TxAmplMan[5:0]							
0x14A8	0x00	<a href="#">RLMSA8[7:0]</a>	FW_PHY _CTRL	FW_PHY _PU_TX	FW_PHY _RSTB	FW_HSK _PHASE	FW_HSK _TYPE	FW_HSK _TX_EN	FW_HSK _RX_EN	FW_CAL IBRATIN G		
0x14A9	0x00	<a href="#">RLMSA9[7:0]</a>	FW_REP CAL_RS TB	FW_STA RT_CAP _CAL	FW_TXD _SQUEL CH	FW_TXD _EN	FW_RX D_EN	FW_AE Q_EN	FW_WB L_EN	FW_HA NDSHAK ED		

ADDRESS	RESET	NAME	MSB							LSB
0x14AA	0x90	<a href="#">RLMSAA[7:0]</a>	RSVD	RSVD	ROR_CL_K_DET	RSVD	RSVD	RSVD	RSVD	RSVD
0x14BA	0x30	<a href="#">RLMSBA[7:0]</a>				TwoDAmpMin[7:0]				
0x14C8	0x7F	<a href="#">RLMSC8[7:0]</a>	–			VmlTxAmpl_reg[6:0]				
0x14C9	0x05	<a href="#">RLMSC9[7:0]</a>	–	–	–	–	mcursor_reg[3:0]			
0x14CA	0x07	<a href="#">RLMSCA[7:0]</a>	–	–	–	–	rcursor_reg[3:0]			
0x14CE	0x01	<a href="#">RLMSCE[7:0]</a>	RSVD	RSVD	RSVD	enminus_reg	enminus_man	RSVD	RSVD	enffe
<b>DPLL REF</b>										
0x1A00	0xF5	<a href="#">DPLL_0[7:0]*</a>	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	config_soft_rst_n
0x1A03	0x82	<a href="#">DPLL_3[7:0]*</a>	config_se_clock_out_use_external	RSVD	RSVD	config_use_intern al_divider_values	config_force_enable_ss	config_spread_bit_ratio[2:0]		
0x1A07	0x04	<a href="#">DPLL_7[7:0]</a>	config_div_fb_L		config_div_in[4:0]			RSVD[1:0]		
0x1A08	0x14	<a href="#">DPLL_8[7:0]</a>			config_div_fb_H[7:0]					
0x1A09	0x40	<a href="#">DPLL_9[7:0]</a>			config_div_out_L[4:0]		config_div_fb_exp[2:0]			
0x1A0A	0x81	<a href="#">DPLL_10[7:0]</a>	config_allow_coarse_change		config_div_out_exp[2:0]		config_div_out_H[3:0]			
<b>EFUSE</b>										
0x1C50	0x00	<a href="#">EFUSE80[7:0]</a>			SERIAL_NUMBER_0[7:0]					
0x1C51	0x00	<a href="#">EFUSE81[7:0]</a>			SERIAL_NUMBER_1[7:0]					
0x1C52	0x00	<a href="#">EFUSE82[7:0]</a>			SERIAL_NUMBER_2[7:0]					
0x1C53	0x00	<a href="#">EFUSE83[7:0]</a>			SERIAL_NUMBER_3[7:0]					
0x1C54	0x00	<a href="#">EFUSE84[7:0]</a>			SERIAL_NUMBER_4[7:0]					
0x1C55	0x00	<a href="#">EFUSE85[7:0]</a>			SERIAL_NUMBER_5[7:0]					
0x1C56	0x00	<a href="#">EFUSE86[7:0]</a>			SERIAL_NUMBER_6[7:0]					
0x1C57	0x00	<a href="#">EFUSE87[7:0]</a>			SERIAL_NUMBER_7[7:0]					
0x1C58	0x00	<a href="#">EFUSE88[7:0]</a>			SERIAL_NUMBER_8[7:0]					
0x1C59	0x00	<a href="#">EFUSE89[7:0]</a>			SERIAL_NUMBER_9[7:0]					
0x1C5A	0x00	<a href="#">EFUSE90[7:0]</a>			SERIAL_NUMBER_10[7:0]					
0x1C5B	0x00	<a href="#">EFUSE91[7:0]</a>			SERIAL_NUMBER_11[7:0]					
0x1C5C	0x00	<a href="#">EFUSE92[7:0]</a>			SERIAL_NUMBER_12[7:0]					
0x1C5D	0x00	<a href="#">EFUSE93[7:0]</a>			SERIAL_NUMBER_13[7:0]					
0x1C5E	0x00	<a href="#">EFUSE94[7:0]</a>			SERIAL_NUMBER_14[7:0]					
0x1C5F	0x00	<a href="#">EFUSE95[7:0]</a>			SERIAL_NUMBER_15[7:0]					
0x1C60	0x00	<a href="#">EFUSE96[7:0]</a>			SERIAL_NUMBER_16[7:0]					
0x1C61	0x00	<a href="#">EFUSE97[7:0]</a>			SERIAL_NUMBER_17[7:0]					
0x1C62	0x00	<a href="#">EFUSE98[7:0]</a>			SERIAL_NUMBER_18[7:0]					
0x1C63	0x00	<a href="#">EFUSE99[7:0]</a>			SERIAL_NUMBER_19[7:0]					
0x1C64	0x00	<a href="#">EFUSE100[7:0]</a>			SERIAL_NUMBER_20[7:0]					

ADDRESS	RESET	NAME	MSB								LSB
0x1C65	0x00	<a href="#">EFUSE101[7:0]</a>									SERIAL_NUMBER_21[7:0]
0x1C66	0x00	<a href="#">EFUSE102[7:0]</a>									SERIAL_NUMBER_22[7:0]
0x1C67	0x00	<a href="#">EFUSE103[7:0]</a>									SERIAL_NUMBER_23[7:0]
<b>FUNC_SAFE</b>											
0x1D00	0x00	<a href="#">REGCRC0[7:0]</a>	-	-	LIMIT_R EG_RAN GE	GEN_R OLLING _CRC	I2C_WR _COMP _UTE	PERIODI C_COM PUTE	CHECK_ CRC	RESET_ CRC	
0x1D01	0x00	<a href="#">REGCRC1[7:0]</a>									CRC_PERIOD[7:0]
0x1D02	0x00	<a href="#">REGCRC2[7:0]</a>									REGCRC LSB[7:0]
0x1D03	0x00	<a href="#">REGCRC3[7:0]</a>									REGCRC_MSB[7:0]
0x1D08	0x00	<a href="#">I2C_UART_C RC0[7:0]</a>	-	-	-	-	-	-	-	RESET_ MSGCN TR	
0x1D09	0x00	<a href="#">I2C_UART_C RC1[7:0]</a>			RSVD[2:0]		RSVD[2:0]			RESET_ MSGCN TR_ERR _CNT	RESET_ CRC_ER R_CNT
0x1D0A	0x00	<a href="#">I2C_UART_C RC2[7:0]</a>									CRC_VAL[7:0]
0x1D0B	0x00	<a href="#">I2C_UART_C RC3[7:0]</a>									MSGCNTR LSB[7:0]
0x1D0C	0x00	<a href="#">I2C_UART_C RC4[7:0]</a>									MSGCNTR_MSB[7:0]
0x1D12	0xE0	<a href="#">FS_INTR0[7:0]</a>	I2C_UA RT_MSG CNTR_E RR_OEN	I2C_UA RT_CRC _ERR_O EN	MEM_E CC_ERR 2_OEN	MEM_E CC_ERR 1_OEN	-	-	-	REG_CR C_ERR_ OEN	
0x1D13	0x00	<a href="#">FS_INTR1[7:0]</a>	I2C_UA RT_MSG CNTR_E RR_INT	I2C_UA RT_CRC _ERR_I NT	MEM_E CC_ERR 2_INT	MEM_E CC_ERR 1_INT	-	-	-	REG_CR C_ERR_ FLAG	
0x1D14	0x00	<a href="#">MEM_ECC0[7:0]</a>			RSVD[2:0]		RSVD[2:0]			RESET_ MEM_E CC_ERR 2_CNT	RESET_ MEM_E CC_ERR 1_CNT
0x1D15	0x00	<a href="#">MEM_ECC1[7:0]</a>									MEM_ECC_ERR1_CNT[7:0]
0x1D16	0x00	<a href="#">MEM_ECC2[7:0]</a>									MEM_ECC_ERR2_CNT[7:0]
0x1D17	0x00	<a href="#">MEM_ECC3[7:0]</a>	-	-	-	-	-	-	MEM_E CC_ERR 2_INJ	MEM_E CC_ERR 1_INJ	
0x1D18	0x00	<a href="#">MEM_ECC4[7:0]</a>									MEM_ECC_ERR_INJ_SEL[7:0]

ADDRESS	RESET	NAME	MSB							LSB
0x1D20	0x00	<a href="#">REG_POST0[7:0]*</a>	POST_D ONE	POST_M BIST_PA SSED	POST_L BIST_PA SSED	-	-	RSVD	RSVD	RSVD
0x1D28	0x00	<a href="#">REGADCBIS T0[7:0]*</a>	RR_ACC URACY	RSVD	-	MUXVE R_EN	-	RUN_AC CURAC Y	RSVD	RUN_TM ON_CAL
0x1D31	0x0F	<a href="#">REGADCBIS T3[7:0]*</a>				REFLIM[7:0]				
0x1D32	0x0F	<a href="#">REGADCBIS T4[7:0]*</a>				REFLIMSCL1[7:0]				
0x1D33	0x07	<a href="#">REGADCBIS T5[7:0]*</a>				REFLIMSCL2[7:0]				
0x1D34	0x07	<a href="#">REGADCBIS T6[7:0]*</a>				REFLIMSCL3[7:0]				
0x1D35	0x03	<a href="#">REGADCBIS T7[7:0]*</a>				TLIMIT[7:0]				
0x1D37	0x00	<a href="#">REGADCBIS T9[7:0]*</a>				MUXV_CTRL[7:0]				
0x1D3A	0xFF	<a href="#">REGADCBIS T12[7:0]*</a>				TMONCAL_OOD_WAIT_B2[7:0]				
0x1D3B	0xFF	<a href="#">REGADCBIS T13[7:0]</a>				T_EST_OUT_B0[7:0]				
0x1D3C	0xC3	<a href="#">REGADCBIS T14[7:0]</a>	T_EST_OUT_B1[1:0]	-	-	-	-	ALT_T_EST_OUT_B1[1:0]		
0x1D3D	0xFF	<a href="#">REGADCBIS T15[7:0]</a>				ALT_T_EST_OUT_B0[7:0]				
0x1D40	0xFF	<a href="#">REGCRC8[7: 0]</a>				SKIP0_LSB[7:0]				
0x1D41	0xFF	<a href="#">REGCRC9[7: 0]</a>				SKIP0_MSB[7:0]				
0x1D42	0xFF	<a href="#">REGCRC10[7: 0]</a>				SKIP1_LSB[7:0]				
0x1D43	0xFF	<a href="#">REGCRC11[7: 0]</a>				SKIP1_MSB[7:0]				
0x1D44	0xFF	<a href="#">REGCRC12[7: 0]</a>				SKIP2_LSB[7:0]				
0x1D45	0xFF	<a href="#">REGCRC13[7: 0]</a>				SKIP2_MSB[7:0]				
0x1D46	0xFF	<a href="#">REGCRC14[7: 0]</a>				SKIP3_LSB[7:0]				
0x1D47	0xFF	<a href="#">REGCRC15[7: 0]</a>				SKIP3_MSB[7:0]				
0x1D48	0xFF	<a href="#">REGCRC16[7: 0]</a>				SKIP4_LSB[7:0]				
0x1D49	0xFF	<a href="#">REGCRC17[7: 0]</a>				SKIP4_MSB[7:0]				
0x1D4A	0xFF	<a href="#">REGCRC18[7: 0]</a>				SKIP5_LSB[7:0]				
0x1D4B	0xFF	<a href="#">REGCRC19[7: 0]</a>				SKIP5_MSB[7:0]				

ADDRESS	RESET	NAME	MSB								LSB
0x1D4C	0xFF	<a href="#">REGCRC20[7:0]</a>									SKIP6_LSB[7:0]
0x1D4D	0xFF	<a href="#">REGCRC21[7:0]</a>									SKIP6_MSB[7:0]
0x1D4E	0xFF	<a href="#">REGCRC22[7:0]</a>									SKIP7_LSB[7:0]
0x1D4F	0xFF	<a href="#">REGCRC23[7:0]</a>									SKIP7_MSB[7:0]
0x1D5F	0x00	<a href="#">CC_RTTN_E RR[7:0]*</a>	-	-	-	-	-	RESET_EFUSE_CRC_ER_R	INJECT_EFUSE_CRC_ER_R	INJECT_RTTN_CRC_ERR	

## Register Details

### [REG0 \(0x0\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
Field	DEV_ADDR[6:0]						CFG_BLOCK			
Reset	0b1000000						0b0			
Access Type	Write, Read						Write, Read			
BITFIELD	BITS	DESCRIPTION				DECODE				
DEV_ADDR	7:1	Device Address Default value is set by the CFG0 pin as follows: CFG0 Device Address 000 0b1000000 001 0b1000010 010 0b1000000 011 0b1000010 100 0b1000010 101 0b1000000 110 0b1000010 111 0b1000000				0b0000000: I <sup>2</sup> C write/read address is 0x00/0x01 0b0000001: I <sup>2</sup> C write/read address is 0x02/0x03 ... ... 0b1000000: I <sup>2</sup> C write/read address is 0x80/0x81 0b1000010: I <sup>2</sup> C write/read address is 0x84/0x85 0b1000100: I <sup>2</sup> C write/read address is 0x88/0x89 0b1100000: I <sup>2</sup> C write/read address is 0xC0/0xC1 0b1100010: I <sup>2</sup> C write/read address is 0xC4/0xC5 0b1100100: I <sup>2</sup> C write/read address is 0xC8/0xC9 0b0100000: I <sup>2</sup> C write/read address is 0x40/0x41 0b0100010: I <sup>2</sup> C write/read address is 0x44/0x45 ... ... 0b1111111: I <sup>2</sup> C write/read address is 0xFE/0xFF				
CFG_BLOCK	0	Configuration Block When set, all registers become non-writable (read-only). This bit can be used to freeze the chip configuration. The only way to clear this register and regain write access is with a power cycle or toggling the PWDNB pin.				0b0: Not Blocked 0b1: Blocked				

[REG1 \(0x1\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	IIC_2_EN	IIC_1_EN	DIS_LOCAL_CC	DIS_Rem_CC	TX_RATE[1:0]		RX_RATE[1:0]			
<b>Reset</b>	0b0	0b0	0b0	0b0	0b10		0b00			
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read		Write, Read			
BITFIELD	BITS	DESCRIPTION			DECODE					
IIC_2_EN	7	Enable pass-through I <sup>2</sup> C Channel 2 (SDA2/RX2, SCL2/TX2)			0b0: I <sup>2</sup> C pass-through Channel 2 disabled 0b1: I <sup>2</sup> C pass-through Channel 2 enabled					
IIC_1_EN	6	Enable pass-through I <sup>2</sup> C Channel 1 (SDA1/RX1, SCL1/TX1)			0b0: I <sup>2</sup> C pass-through Channel 1 disabled 0b1: I <sup>2</sup> C pass-through Channel 1 enabled					
DIS_LOCAL_CC	5	Disable 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 channel					
DIS_Rem_CC	4	Disable access to remote device control-channel over GMSL connection			0b0: Remote control channel enabled 0b1: Remote control channel disabled					
TX_RATE	3:2	Transmitter (forward channel) bit rate (when changed, becomes active after next link reset).  Default value is set by the configuration pins at power-up.			0b00: Reserved 0b01: 3Gbps 0b10: 6Gbps 0b11: Reserved					
RX_RATE	1:0	Receiver (reverse channel) bit rate (when changed, becomes active after next link reset)			0b00: 187.5Mbps 0b01: Reserved 0b10: Reserved 0b11: Reserved					

[REG2 \(0x2\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	VID_TX_EN_Z	—	—	—	—	INFO_TX_EN	INFO_RX_EN
<b>Reset</b>	—	0b1	—	—	—	—	0b1	0b1
<b>Access Type</b>	—	Write, Read	—	—	—	—	Write, Read	Write, Read
BITFIELD	BITS	DESCRIPTION			DECODE			
VID_TX_EN_Z	6	Video Transmit Enable for Video Pipe Z			0b0: Video transmit Pipe Z disabled 0b1: Video transmit Pipe Z enabled			
INFO_TX_EN	1	Enable Info Frame transmission						
INFO_RX_EN	0	Enable processing of incoming Info Frames						

[REG3 \(0x3\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	RSVD	UART_1_EN	–	RCLK_ALT	RCLKSEL[1:0]	
<b>Reset</b>	–	–	0b0	0b0	–	0b0	0b00	
<b>Access Type</b>	–	–		Write, Read	–	Write, Read	Write, Read	
<b>BITFIELD</b>	<b>BITS</b>	<b>DESCRIPTION</b>					<b>DECODE</b>	
UART_1_EN	4	Enable pass-through UART Channel 1 (SDA1/RX1, SCL1/TX1)					0b0: Pass-through UART Channel 1 disabled 0b1: Pass-through UART Channel 1 enabled	
RCLK_ALT	2	Selects MFP pin to route RCLK to.					0b0: Route RCLK to MFP4 0b1: Route RCLK to MFP2 (alternate RCLK output pin)	
RCLKSEL	1:0	RCLKOUT clock selection					0b00: XTAL/1 = 25MHz 0b01: XTAL/2 = 12.5MHz 0b10: XTAL/4 = 6.25MHz 0b11: Reference PLL output	

[REG4 \(0x4\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	–	CC_MSGC_NTR_EN	CC_CRC_EN	CC_CRC_MSGCNTR_OVR	RSVD	XTAL_PU
<b>Reset</b>	–	–	–	0b1	0b1	0b0	0b0	0b0
<b>Access Type</b>	–	–	–	Write, Read	Write, Read	Write, Read		Write, Read
<b>BITFIELD</b>	<b>BITS</b>	<b>DESCRIPTION</b>					<b>DECODE</b>	
CC_MSGCNTR_EN	4	Enable I <sup>2</sup> C/UART message counter override when set to 1. Only active when CC_CRC_MSGCNTR_OVR = 1					0x0: I <sup>2</sup> C/UART Message Counter not enabled 0x1: I <sup>2</sup> C/UART Message Counter enabled	
CC_CRC_EN	3	Enable I <sup>2</sup> C/UART CRC override when set to 1. Only active when CC_CRC_MSGCNTR_OVR = 1					0b0: I <sup>2</sup> C/UART CRC not enabled 0b1: I <sup>2</sup> C/UART CRC enabled	
CC_CRC_MSGCNTR_OVR	2	Enable manual override of I <sup>2</sup> C/UART CRC or message counter configuration when set to 1. When 0, use default CRC and message counter configuration.					0b0: Default CRC and Message Counter configuration 0b1: Manual override of CRC and Message Counter configuration. CC_CRC_EN and CC_MSGCNTR_EN register bits control I <sup>2</sup> C / UART CRC and Message Counter options respectively.	
XTAL_PU	0	Enable XTAL as reference clock. The initial value of this register is set by the CFG pin selection. Writing to this register bit overrides CFG pin.					0b0: Reverse Channel Reference (ROR) used as reference clock 0b1: XTAL used as reference clock	

**REG5 (0x5)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	LOCK_EN	ERRB_EN	ALT_LOCK_EN	ALT_ERRB_EN	RSVD	RSVD	PU_LF1	PU_LF0
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read			Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
LOCK_EN	7	Enable LOCK Output	0b0: LOCK output disabled 0b1: LOCK output enabled
ERRB_EN	6	Enable ERRB Output	0b0: ERRB output disabled 0b1: ERRB output enabled
ALT_LOCK_EN	5	Enable LOCK output on alternate output	0b0: LOCK output disabled 0b1: LOCK output enabled
ALT_ERRB_EN	4	Enable ERRB output on alternate output	0b0: ERRB output disabled 0b1: ERRB output enabled
PU_LF1	1	Power Up Line-Fault Monitor 1	0b0: Line-fault monitor 1 disabled 0b1: Line-fault monitor 1 enabled
PU_LF0	0	Power Up Line-Fault Monitor 0	0b0: Line-fault monitor 0 disabled 0b1: Line-fault monitor 0 enabled

**REG6 (0x6)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	—	RCLKEN	I2CSEL	—	—	—	RSVD
<b>Reset</b>	0b1	—	0b0	0b0	—	—	—	0b0
<b>Access Type</b>		—	Write, Read	Write, Read	—	—	—	

BITFIELD	BITS	DESCRIPTION	DECODE
RCLKEN	5	Enable/disable RCLK Output.	0b0: RCLK output is disabled 0b1: RCLK output is enabled
I2CSEL	4	I <sup>2</sup> C/UART selection Bit is set according to the CFG0 pin value at power-up. Changing this value through a register write is not recommended. Instead, change the CFG0 value and power up the device again.	0b0: UART 0b1: I <sup>2</sup> C

**REG7 (0x7)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD[3:0]				RSVD	RSVD	RSVD	PAR_VID_E_N
<b>Reset</b>	0xF				0b0	0b1	0b1	0b0
<b>Access Type</b>								Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
PAR_VID_EN	0	Enable parallel video input	0b0: Parallel video disabled 0b1: Parallel video enabled

REG13 (0xD)

BIT	7	6	5	4	3	2	1	0
Field	DEV_ID[7:0]							
Reset	0xB7							
Access Type	Read Only							
BITFIELD	BITS	DESCRIPTION					DECODE	
DEV_ID	7:0	Device Identifier					0xBF: MAX96717	

REG14 (0xE)

BIT	7	6	5	4	3	2	1	0
Field	RSVD[3:0]							
Reset	0x0							
Access Type	Read Only							
BITFIELD	BITS	DESCRIPTION					DECODE	
DEV_REV	3:0	Device Revision					0xX: Device revision number (RevID)	

REG15 (0xF)

BIT	7	6	5	4	3	2	1	0
Field	RSVD	RSVD	SPEED_CPBL[1:0]		RSVD	RSVD	RSVD	RSVD
Reset	0b0	0b0	0b00		0b0	0b0	0b0	0b0
Access Type			Read Only					
BITFIELD	BITS		DESCRIPTION					
SPEED_CPBL	5:4		Video resolution capability					

REG26 (0x26)

BIT	7	6	5	4	3	2	1	0
Field	—	LF_1[2:0]			—	LF_0[2:0]		
Reset	—	0b010			—	0b010		
Access Type	—	Read Only			—	Read Only		
BITFIELD	BITS	DESCRIPTION					DECODE	
LF_1	6:4	Line-fault status of wire connected to LMN1 pin					0b000: Short to battery 0b001: Short to GND 0b010: Normal Operation 0b011: Line Open 0b1XX: Line-to-line short	

BITFIELD	BITS	DESCRIPTION	DECODE
LF_0	2:0	Line-fault status of wire connected to LMN0 pin	0b000: Short to battery 0b001: Short to GND 0b010: Normal Operation 0b011: Line Open 0b1XX: Line-to-line short

PWR0 (0x8)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VDDBAD_STATUS[2:0]						CMP_STATUS[4:0]	
<b>Reset</b>	0b000						0b00000	
<b>Access Type</b>	Read Only						Read Only	

BITFIELD	BITS	DESCRIPTION	DECODE
VDDBAD_STATUS	7:5	Switched 1V supply comparator status bits Note that when CAP_VDD < 0.82V, the device is operating outside of the specified conditions and usually powers down.	0bXX1: Latched high when CAP_VDD < 0.82V 0bX1X: Latched high when CAP_VDD < 0.82V 0b1XX: Reserved
CMP_STATUS	4:0	V <sub>DD18</sub> , V <sub>DDIO</sub> , and CAP_VDD supply voltage comparator status bits	0bXXXX0: Latched low when V <sub>DD18</sub> < 1.617V 0bXXX0X: Latched low when switched V <sub>DDIO</sub> supply < 1.617V 0bXX0XX: Latched low when CAP_VDD < 0.82V 0bX0XXX: Reserved 0b0XXXX: Reserved 0bXX111: All supplies are at expected levels

PWR1 (0x9)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVERTEMP	VDD_OV_LIVE	PORZ_STATUS[5:0]					
<b>Reset</b>	0b0	0b0	0b000000					
<b>Access Type</b>	Read Only	Read Only	Read Only					

BITFIELD	BITS	DESCRIPTION	DECODE
OVERTEMP	7	Temperature monitor overtemperature indicator. Trip temperature Ttrip set by SET_TEMP1P0[1:0] in CMU6	0x0: T < Ttrip 0x1: T > Ttrip
VDD_OV_LIVE	6	Live V <sub>DDSW</sub> overvoltage status V <sub>DDSW</sub> level set by OV_LEVEL in register PM_OV_STAT	0x0: V <sub>DDSW</sub> < V <sub>TRIP_OV</sub> 0x1: V <sub>DDSW</sub> > V <sub>TRIP_OV</sub>
PORZ_STATUS	5:0	PORZ status Bit 0: Latched low when either V <sub>DD18</sub> < 1.516V or V <sub>DDIO</sub> < 1.055V Bit 1: Reserved Bit 2: Latched low when V <sub>DDIO</sub> < 1.055V Bit 3: Reserved Bit 4: Latched low when V <sub>DD18</sub> < 1.516V (resets 1V domain, need to raise V <sub>DD18</sub> above 1.516V to read registers) Bit 5: Reserved	

PWR4 (0xC)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	DIS_LOCAL_WAKE	—	WAKE_EN_A	RSVD[3:0]			
<b>Reset</b>	0b0	0b0	—	0b1	0x5			
<b>Access Type</b>		Write, Read	—	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
DIS_LOCAL_WAKE	6	Disable wake-up by local µC from SDA_RX pin				0b0: Local wake-up enabled 0b1: Local wake-up disabled		
WAKE_EN_A	4	Enable wake-up by remote chip connected to GMSL Link				0b0: GMSL Link remote wake-up disabled 0b1: GMSL Link remote wake-up enabled		

CTRL0 (0x10)\*

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	RESET_ALL	RESET_LINK	RESET_ONESHOT	RSVD	SLEEP	—	RSVD[1:0]			
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	—	0b01			
<b>Access Type</b>	Write Clears All, Read	Write, Read	Write Clears All, Read	—	Write, Read	—				
BITFIELD	BITS	DESCRIPTION				DECODE				
RESET_ALL	7	Writing 1 to this bit resets the device, including all blocks. Registers are reset to defaults. This is equivalent to toggling the PWDNB pin. The bit is cleared when written.				0b0: No action 0b1: Activate chip reset				
RESET_LINK	6	Reset data path (keep register settings). Write 1 to activate reset. Write 0 to release reset.				0b0: Release link reset 0b1: Activate link reset				
RESET_ONESHOT	5	Reset data path (keep register settings). Write 1 to activate reset, bit self clears and automatically releases reset.				0b0: No action 0b1: Reset data path				
SLEEP	3	Activate Sleep Mode				0b0: Sleep mode disabled 0b1: Sleep mode enabled				

CTRL1 (0x11)\*

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	RSVD	RSVD	RSVD	—	—	—	RSVD	CXTP_A		
<b>Reset</b>	0b0	0b0	0b0	—	—	—	0b1	0b0		
<b>Access Type</b>				—	—	—	Write, Read			
BITFIELD	BITS	DESCRIPTION				DECODE				
CXTP_A	0	Coax/Twisted-pair cable select for GMSL Link Bit is set according to the latched CFG1 pin value at power-up				0b0: Shielded twisted-pair drive 0b1: Coax drive				

CTRL2 (0x12)

BIT	7	6	5	4	3	2	1	0				
<b>Field</b>	RSVD	RSVD	—	LDO_BYPASS	RSVD[1:0]		RSVD[1:0]					
<b>Reset</b>	0b0	0b0	—	0b0	0b01		0b00					
<b>Access Type</b>			—	Write, Read								
BITFIELD	BITS		DESCRIPTION									
LDO_BYPASS	4		Enable LDO bypass									

CTRL3 (0x13)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	RSVD	RSVD[1:0]		LOCKED	ERROR	CMU_LOCKED	—
<b>Reset</b>	0b0	0b0	0b01		0b0	0b0	0b0	—
<b>Access Type</b>					Read Only	Read Only	Read Only	—
BITFIELD	BITS		DESCRIPTION			DECODE		
LOCKED	3		GMSL Link Locked (bidirectional)			0b0: GMSL link not locked 0b1: GMSL link locked		
ERROR	2		Reflects global error status			0b0: ERRB not asserted (ERRB pin = 1) 0b1: ERRB asserted (ERRB pin = 0)		
CMU_LOCKED	1		Clock Multiplier Unit (CMU) Locked			0b0: CMU not locked 0b1: CMU locked		

INTR0 (0x18)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	RSVD	RSVD	—	AUTO_ERR_RST_EN	DEC_ERR_THR[2:0]		
<b>Reset</b>	0b1	0b0	0b1	—	0b0	0b000		
<b>Access Type</b>				—	Write, Read	Write, Read		
BITFIELD	BITS		DESCRIPTION			DECODE		
AUTO_ERR_RST_EN	3		Automatically resets DEC_ERR_A (0x22) and IDLE_ERR (0x24) bitfields after ERRB pin is asserted for 1µs.			0b0: Auto reset disabled 0b1: Auto reset enabled		
DEC_ERR_THR	2:0		Decoding and idle-error reporting threshold. Threshold controls the number of errors that can happen before the error flags assert. DEC_ERR_FLAG_A is asserted when DEC_ERR_A ≥ DEC_ERR_THR IDLE_ERR_FLAG is asserted when IDLE_ERR ≥ DEC_ERR_THR			0b000: 1 0b001: 2 0b010: 4 0b011: 8 0b100: 16 0b101: 32 0b110: 64 0b111: 128		

[INTR1 \(0x19\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PKT_CNT_EXP[3:0]					AUTO_CNT_RST_EN	RSVD[2:0]	
<b>Reset</b>	0x0					0b0	0b000	
<b>Access Type</b>	Write, Read					Write, Read		
BITFIELD	BITS	DESCRIPTION					DECODE	
PKT_CNT_EXP	7:4	Packet Count Multiplier Exponent See the description of PKT_CNT (0x25) bitfield.					0bXXX: PKT_CNT exponent	
AUTO_CNT_RST_EN	3	Automatically reset PKT_CNT (0x25) bitfield after ERRB pin is asserted for 1µs.					0b0: Auto reset disabled 0b1: Auto reset enabled	

[INTR2 \(0x1A\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFGEN_UNLOCKED_OEN	RSVD	REM_ERR_OEN	—	LFLT_INT_OEN	IDLE_ERR_OEN	—	DEC_ERR_OEN_A
<b>Reset</b>	0b0	0b0	0b0	—	0b1	0b0	—	0b1
<b>Access Type</b>	Write, Read		Write, Read	—	Write, Read	Write, Read	—	Write, Read
BITFIELD	BITS	DESCRIPTION					DECODE	
REFGEN_UNLOCKED_OEN	7	Enable reporting of reference clock DPLL not locked					0b0: Reporting disabled 0b1: Reporting enabled	
REM_ERR_OEN	5	Enable reporting of remote error status (REM_ERR - 0x1B) at ERRB pin					0b0: Reporting disabled 0b1: Reporting enabled	
LFLT_INT_OEN	3	Enable reporting of line-fault interrupt (LFLT_INT - 0x1B) at ERRB pin					0b0: Reporting disabled 0b1: Reporting enabled	
IDLE_ERR_OEN	2	Enable reporting of idle-word errors (IDLE_ERR_FLAG - 0x1B) at ERRB pin					0b0: Reporting disabled 0b1: Reporting enabled	
DEC_ERR_OEN_A	0	Enable reporting of decoding errors (DEC_ERR_FLAG_A - 0x1B) at ERRB pin					0b0: Reporting disabled 0b1: Reporting enabled	

[INTR3 \(0x1B\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFGEN_UNLOCKED	RSVD	REM_ERR_FLAG	—	LFLT_INT	IDLE_ERR_FLAG	—	DEC_ERR_FLAG_A
<b>Reset</b>	0b0	0b0	0b0	—	0b0	0b0	—	0b0
<b>Access Type</b>	Read Only		Read Only	—	Read Only	Read Only	—	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
REFGEN_U_NLOCKED	7	Reference DPLL generating RCLKOUT is not locked. This bit is only valid when the reference DPLL is enabled and should be ignored when it is not.  See register 0x3F0.	0b0: REFGEN DPLL is locked 0b1: REFGEN DPLL not locked
REM_ERR_F_LAG	5	Received remote side error status (inverse of remote side ERRB pin level)	0b0: No remote side error 0b1: Remote side error
LFLT_INT	3	Line-Fault Interrupt  Asserted when either one of line-fault monitors indicates a fault status.  See LF_0 (0x26) and LF_1 (0x26) bitfields for more information.	0b0: No line fault 0b1: Line fault
IDLE_ERR_F_LAG	2	Idle-Word Error Flag  Asserted when IDLE_ERR (0x24) $\geq$ DEC_ERR_THR (0x18).	0b0: Flag not asserted 0b1: Flag asserted
DEC_ERR_F_LAG_A	0	GMSL Packet Decoding Error Flag Asserted when DEC_ERR_A (at addr 0x22) $\geq$ DEC_ERR_THR (at addr 0x18).	0b0: Error flag not asserted 0b1: Error flag asserted

INTR4 (0x1C)\*

BIT	7	6	5	4	3	2	1	0
Field	VREG_OV_OEN	EOM_ERR_OEN_A	VDD_OV_OEN	VDD18_OV_OEN	MAX_RT_OEN	RT_CNT_OEN	PKT_CNT_OEN	—
Reset	0b0	0b0	0b0	0b0	0b1	0b0	0b0	—
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	—

BITFIELD	BITS	DESCRIPTION	DECODE
VREG_OV_OEN	7	Enable CAP_VDD overvoltage status on ERRB	0b0: Reporting disabled 0b1: Reporting enabled
EOM_ERR_OEN_A	6	Enable reporting of eye-opening monitor error (EOM_ERR_FLAG_A - 0x1D) for GMSL Link at ERRB pin	0b0: Reporting disabled 0b1: Reporting enabled
VDD_OV_OEN	5	Enable V <sub>DD</sub> overvoltage status on ERRB	0b0: Reporting disabled 0b1: Reporting enabled
VDD18_OV_OEN	4	Enable V <sub>DD18</sub> overvoltage status on ERRB	0b0: Reporting disabled 0b1: Reporting enabled
MAX_RT_OEN	3	Enable reporting of combined ARQ maximum retransmission limit error flag (MAX_RT_FLAG - 0x1D) at ERRB pin	0b0: Reporting disabled 0b1: Reporting enabled
RT_CNT_OEN	2	Enable reporting of combined ARQ retransmission event flag (RT_CNT_FLAG - 0x1C) at ERRB pin	0b0: Reporting disabled 0b1: Reporting enabled
PKT_CNT_OEN	1	Enable reporting of packet count flag (PKT_CNT_FLAG - 0x1E) at ERRB pin	0b0: Reporting disabled 0b1: Reporting enabled

INTR5 (0x1D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VREG_OV_FLAG	EOM_ERR_FLAG_A	VDD_OV_FLAG	VDD18_OV_FLAG	MAX_RT_FLAG	RT_CNT_FLAG	PKT_CNT_FLAG	—
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	—
<b>Access Type</b>	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Only	Read Only	Read Only	—

BITFIELD	BITS	DESCRIPTION	DECODE
VREG_OV_FLAG	7	CAP_VDD Overvoltage Indication  This bit is sticky. It is set when CAP_VDD is over the overvoltage threshold. It is cleared when read.	0b0: Flag not asserted 0b1: Flag asserted
EOM_ERR_FLAG_A	6	Eye-opening is below the configured threshold set by EOM_MIN_THR[6:0]. This bit is sticky. It is cleared when read.	0b0: Flag not asserted 0b1: Flag asserted
VDD_OV_FLAG	5	V <sub>DD</sub> Overvoltage Indication  This bit is sticky. It is set when V <sub>DD</sub> is over the overvoltage threshold. It is cleared when read.	0b0: Flag not asserted 0b1: Flag asserted
VDD18_OV_FLAG	4	V <sub>DD18</sub> Overvoltage Flag  This bit is sticky. It is set when V <sub>DD18</sub> is over the overvoltage threshold. It is cleared when read.	0b0: Flag not asserted 0b1: Flag asserted
MAX_RT_FLAG	3	Combined ARQ maximum retransmission limit error flag  Asserted when any of the selected channel's ARQ retransmission limit is reached. Selection is done by each channel's MAX_RT_ERR_OEN (0x1C) bitfield.	0b0: Flag not asserted 0b1: Flag asserted
RT_CNT_FLAG	2	Combined ARQ retransmission event flag  Asserted when any of the selected channels have done at least one ARQ retransmission. Selection is done by each channel's RT_CNT_OEN (0x1C) bitfield.	0b0: Flag not asserted 0b1: Flag asserted
PKT_CNT_FLAG	1	Packet Count Flag  Asserted when PKT_CNT (0x25) ≥ PKT_CNT_THR (0x19)	0b0: Flag not asserted 0b1: Flag asserted

[INTR6 \(0x1E\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VDDCMP_IN_T_OEN	PORZ_INT_OEN	VDDBAD_IN_T_OEN	EFUSE_CRC_ERR_OEN	RTTN_CRC_ERR_OEN	ADC_INT_OEN	RSVD	MIPI_ERR_OEN
<b>Reset</b>	0b1	0b1	0b1	0b1	0b1	0b0	0b1	0b1
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read		Write, Read
BITFIELD	BITS	<b>DESCRIPTION</b>					<b>DECODE</b>	
VDDCMP_IN_T_OEN	7	Enable reporting of combined undervoltage comparator output (VDDCMP_INT_FLAG) at ERRB pin					0x0: Disable reporting 0x1: Enable reporting	
PORZ_INT_OEN	6	Enable reporting of PORZ interrupt (PORZ_INT_FLAG - 0x1F) at ERRB pin. PORZ is monitoring of undervoltage levels of V <sub>DD18</sub> and V <sub>DDIO</sub> .					0x0: Disable reporting 0x1: Enable reporting	
VDDBAD_IN_T_OEN	5	Enable reporting of V <sub>DDBAD</sub> interrupt (VDDBAD_INT_FLAG) at ERRB pin					0x0: Disable reporting 0x1: Enable reporting	
EFUSE_CRC_ERR_OEN	4	Enable reporting eFuse CRC at ERRB pin					0x0: Disable reporting 0x1: Enable reporting	
RTTN_CRC_ERR_OEN	3	Enable reporting of CRC errors (RTTN_CRC_ERR) incurred during readout of retention memory during exit of sleep mode at ERRB pin					0x0: Disable reporting 0x1: Enable reporting	
ADC_INT_OEN	2	Enable reporting of ADC interrupts (ADC_INT_FLAG) at ERRB pin. Individual interrupts also have individual enable bits. See registers ADC_INTRIE0-3.					0x0: Disable reporting 0x1: Enable reporting	
MIPI_ERR_OEN	0	Enable reporting of MIPI RX errors (MIPI_ERR_FLAG) at ERRB pin					0b0: Reporting disabled 0b1: Reporting enabled	

[INTR7 \(0x1F\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VDDCMP_IN_T_FLAG	PORZ_INT_FLAG	VDDBAD_IN_T_FLAG	EFUSE_CRC_ERR	RTTN_CRC_INT	ADC_INT_FLAG	RSVD	MIPI_ERR_FLAG
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Read Clears All	Read Clears All	Read Clears All	Read Only	Read Only	Read Only		Read Only
BITFIELD	BITS	<b>DESCRIPTION</b>					<b>DECODE</b>	
VDDCMP_IN_T_FLAG	7	Combined undervoltage comparator output Asserts when a bit in the CMP_STATUS register is asserted.					0b0: Flag not asserted 0b1: Flag asserted	
PORZ_INT_FLAG	6	PORZ interrupt flag. PORZ is monitoring of undervoltage levels of V <sub>DD18</sub> and V <sub>DDIO</sub> . Asserts when V <sub>DD18</sub> < 1.516V or V <sub>DDIO</sub> < 1.055V.					0b0: Flag not asserted 0b1: Flag asserted	

BITFIELD	BITS	DESCRIPTION	DECODE
VDDBAD_IN_T_FLAG	5	Combined $V_{DD}$ bad indicator. Asserts when either VDDBAD_STATUS[1] or [0] bits is a 1, indicating that CAP_VDD (switched digital core $V_{DD}$ ) has fallen below 0.82V.	0x0: CAP_VDD has not fallen below 0.82V 0x1: CAP_VDD has fallen below 0.82V at least once since this bit is last read
EFUSE_CRC_ERR	4	eFuse CRC error indicator	0b0: Flag not asserted 0b1: Flag asserted
RTTN_CRC_INT	3	Retention memory restore CRC error interrupt. When the device wakes up, contents of retention memory is loaded back to main registers. The restored data is covered by CRC. If CRC fails, this bit is set.	0b0: Flag not asserted 0b1: Flag asserted
ADC_INT_FLAG	2	ADC Interrupt. Individual interrupts also have individual status flags. See registers ADC_INTR0-3.	0b0: Flag not asserted 0b1: Flag asserted
MIPI_ERR_FLAG	0	MIPI RX error flag, asserted when any of these is asserted: phy1_hs_err [0],[1],[4],[5] phy2_hs_err [0],[1],[4],[5] ctrl1_csi_err_l [0],[1],[7] ctrl1_csi_err_h [0]	0b0: Flag not asserted 0b1: Flag asserted

INTR8 (0x20)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ERR_TX_EN	–	–	ERR_TX_ID[4:0]				
<b>Reset</b>	0b1	–	–	0b11111				
<b>Access Type</b>	Write, Read	–	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
ERR_TX_EN	7	Transmit local error status (inverse of ERRB pin level) to remote side through GPIO channel	0b0: Transmit error status disabled 0b1: Transmit error status enabled
ERR_TX_ID	4:0	GPIO ID used for transmitting ERR_TX. Error status is transmitted to remote side using the GPIO interface, using this value as a special GPIO index. It is not recommended to change this value.	0bXXXXX: Value of GPIO ID for transmitting ERR_TX

INTR9 (0x21)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ERR_RX_EN	RSVD	–	ERR_RX_ID[4:0]				
<b>Reset</b>	0b1	0b1	–	0b11111				
<b>Access Type</b>	Write, Read		–	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
ERR_RX_EN	7	Enable reception of remote error status (inverse of ERRB pin level) through GPIO channel			0b0: Receive error status disabled 0b1: Receive error status enabled			

BITFIELD	BITS	DESCRIPTION	DECODE
ERR_RX_ID	4:0	GPIO ID used for receiving ERR_RX. Error status is received from remote side using the GPIO interface, using this value as a special GPIO index. It is not recommended to change this value.	0bXXXXX: Value of GPIO ID for receiving ERR_TX

[CNT0 \(0x22\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DEC_ERR_A[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Clears All							

BITFIELD	BITS	DESCRIPTION	DECODE
DEC_ERR_A	7:0	Number of decoding (disparity) errors detected in packets received over GMSL link. Cleared on read or upon link transitioning from unlock to lock state.	0xXX: Number of detected GMSL link disparity errors

[CNT2 \(0x24\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	IDLE_ERR[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Clears All							

BITFIELD	BITS	DESCRIPTION	DECODE
IDLE_ERR	7:0	Number of idle-word errors detected. Reset after reading or with the rising edge of LOCK.	0xXX: Number of idle-word errors detected

[CNT3 \(0x25\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PKT_CNT[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Clears All							

BITFIELD	BITS	DESCRIPTION	DECODE
PKT_CNT	7:0	<p>Number of received packets of a selected type.</p> <p>Packet type is selected with PKT_CNT_SEL (0x2C register).</p> <p>Reported packet count is a scaled value, such that actual packet count is <math>\geq</math> PKT_CNT <math>\times</math> (2<sup>PKT_CNT_EXP</sup>) and <math>&lt;</math> (PKT_CNT + 1) <math>\times</math> (2<sup>PKT_CNT_EXP</sup>).</p> <p>When maximum value is reported, packet count is greater or equal to the reported value.</p>	0xXX: Scaled number of received packets

[TX0 \(0x28\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD[1:0]		RSVD	RSVD	–	–	TX_FEC_EN	–
<b>Reset</b>	0b01		0b1	0b0	–	–	0x0	–
<b>Access Type</b>					–	–	Write, Read	–

BITFIELD	BITS	DESCRIPTION	DECODE
TX_FEC_EN	1	<p>Enable Forward-Error Correction (FEC) in forward direction.</p> <p>When the part is started up in GMSL2 mode, FEC is default off, and this bit is 0.</p> <p>For GMSL3 parts, if a part is started up in GMSL3 mode, FEC is on by default, and this bit is 1.</p>	0x0: FEC off 0x1: FEC on

[TX1 \(0x29\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	LINK_PRBS_GEN	RSVD	–	ERRG_EN_A	TX_FEC_CR_EN	–	DIS_SCR	DIS_ENC
<b>Reset</b>	0b0	0x0	–	0b0	0x1	–	0b0	0b0
<b>Access Type</b>	Write, Read		–	Write, Read	Write, Read	–	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
LINK_PRBS_GEN	7	Enable link PRBS-7 generator	0x0: Disabled 0x1: Enabled
ERRG_EN_A	4	Error Generator Enable for GMSL Link. Error injection applies to all data going across the link.	0b0: GMSL Link error generator disabled 0b1: GMSL Link error generator enabled
TX_FEC_CR_EN	3	<p>Enable CRC at each FEC block</p> <p>Setting must match deserializer. It is not recommended to change this value.</p>	0x0: Off 0x1: On
DIS_SCR	1	Disable scrambler	
DIS_ENC	0	Disable 9b10b encoding	

[TX2 \(0x2A\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	ERRG_CNT[1:0]		ERRG_RATE[1:0]		ERRG_BURST[2:0]			ERRG_PER		
<b>Reset</b>	0b00		0b10		0b000			0b0		
<b>Access Type</b>	Write, Read		Write, Read		Write, Read			Write, Read		
BITFIELD	BITS	DESCRIPTION				DECODE				
ERRG_CNT	7:6	Number of errors to be generated				0b00: Continuous 0b01: 16 0b10: 128 0b11: 1024				
ERRG_RATE	5:4	Error generator average bit error rate				0b00: 1 in 5120 bits 0b01: 1 in 81920 bits 0b10: 1 in 1310720 bits 0b11: 1 in 20971520 bits				
ERRG_BURST	3:1	Error generator burst error length in bits				0b000: 1 0b001: 2 0b010: 3 0b011: 4 0b100: 8 0b101: 12 0b110: 16 0b111: 20				
ERRG_PER	0	Error generator error distribution selection				0b0: Pseudorandom 0b1: Periodic				

[TX3 \(0x2B\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	RSVD[1:0]		TX_FEC_A CTIVE	—	—	RSVD[2:0]				
<b>Reset</b>	0b01		—		0b100					
<b>Access Type</b>			Read Only		—					
BITFIELD	BITS	DESCRIPTION				DECODE				
TX_FEC_ACTIVE	5	Indicates FEC is active Communication with deserializer must have been established for this bit to get set by hardware.				0x0: Not active 0x1: Actively generating error checking bits.				

[RX0 \(0x2C\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PKT_CNT_LBW[1:0]		—	RSVD	PKT_CNT_SEL[3:0]			
<b>Reset</b>	0b00		—	0b0	0x0			
<b>Access Type</b>	Write, Read		—	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
PKT_CNT_L_BW	7:6	Select the subtype of low-bandwidth packets to count at PKT_CNT (0x25) bitfield	0b00: Count low-bandwidth data packets only 0b01: Count low-bandwidth acknowledge packets only 0b10: Count low-bandwidth data and acknowledge packets 0b11: Reserved
PKT_CNT_SEL	3:0	Select the type of received packets to count at PKT_CNT (0x25) bitfield	0x0: None 0x1: VIDEO 0x2: Reserved 0x3: INFO Frame 0x4: SPI 0x5: I2C 0x6: UART 0x7: GPIO 0x8: Reserved 0x9: Reserved 0xA: Reserved 0xB: Reserved 0xC: Reserved 0xD: Reserved 0xE: All 0xF: Unknown and packets with error

[RX1 \(0x2D\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	LINK_PRBS_CHK	—	RSVD[1:0]		RSVD[1:0]		RSVD	RSVD	
<b>Reset</b>	0b0	—	0b10		0b10		0b0	0b0	
<b>Access Type</b>	Write, Read	—							
BITFIELD	BITS	DESCRIPTION				DECODE			
LINK_PRBS_CHK	7	Enable link PRBS-7 checker				0x0: Disabled 0x1: Enabled			

[GPIOA \(0x30\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	RSVD	GPIO_FWD_CDLY[5:0]					
<b>Reset</b>	0b0	0b1	0b000001					
<b>Access Type</b>			Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_FWD_CDLY	5:0	<p>Compensation delay multiplier for the forward direction.</p> <p>This must be the same value as GPIO_FWD_CDLY of the chip on the other side of the link.</p> <p>Total delay is the (value + 1) multiplied by 1.7<math>\mu</math>s. Default delay is 3.4<math>\mu</math>s.</p> <p>See the GMSL user guide for further information.</p>	0bXXXXXX: Forward compensation delay multiplier value

[GPIOB \(0x31\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD[1:0]		GPIO_REV_CDLY[5:0]					
<b>Reset</b>	0b10		0b001000					
<b>Access Type</b>			Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_REV_CDLY	5:0	<p>Compensation delay multiplier for the reverse direction.</p> <p>This must be the same value as GPIO_REV_CDLY of the chip on the other side of the link.</p> <p>Total delay is the (value + 1) multiplied by 1.7<math>\mu</math>s. Default delay is 15.3<math>\mu</math>s.</p> <p>See the GMSL user guide for further information.</p>	0bXXXXXX: Reverse compensation delay multiplier value

[I2C\\_0 \(0x40\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	SLV_SH[1:0]			SLV_TO[2:0]		
<b>Reset</b>	—	—	0b10			0b110		
<b>Access Type</b>	—	—	Write, Read			Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
SLV_SH	5:4	<p>I<sup>2</sup>C-to-I<sup>2</sup>C subordinate-setup and hold-time setting.</p> <p>Configures the interval between SDA and SCL transitions when driven by the internal I<sup>2</sup>C subordinate.</p>	<p>0b00: Set for I<sup>2</sup>C Fast-mode Plus speed (1Mbps)</p> <p>0b01: Set for I<sup>2</sup>C Fast-mode speed (400Kbps)</p> <p>0b10: Set for I<sup>2</sup>C Standard-mode speed (100Kbps)</p> <p>0b11: Reserved</p>

BITFIELD	BITS	DESCRIPTION	DECODE
SLV_TO	2:0	I <sup>2</sup> C-to-I <sup>2</sup> C subordinate timeout setting.  Internal GMSL2 I <sup>2</sup> C subordinate times out after the configured duration if it does not receive any response while waiting for a packet from remote device.	0b000: 16µs 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled

I<sup>2</sup>C\_1 (0x41)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD		MST_BT[2:0]		–		MST_TO[2:0]	
<b>Reset</b>	0b0		0b101		–		0b110	
<b>Access Type</b>			Write, Read		–		Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
MST_BT	6:4	I <sup>2</sup> C-to-I <sup>2</sup> C main bit rate setting  Configures the I <sup>2</sup> C bit rate used by the internal I <sup>2</sup> C main (in the device on the remote side from the external I <sup>2</sup> C main) Set this according to the I <sup>2</sup> C speed mode.	0b000: 9.92Kbps - Set for I <sup>2</sup> C Standard mode speed 0b001: 33.2Kbps - Set for I <sup>2</sup> C Standard mode speed 0b010: 99.2Kbps - Set for I <sup>2</sup> C Standard or Fast-mode speed 0b011: 123Kbps - Set for I <sup>2</sup> C Fast-mode speed 0b100: 203Kbps - Set for I <sup>2</sup> C Fast-mode speed 0b101: 397Kbps - Set for I <sup>2</sup> C Fast or Fast-mode Plus speed 0b110: 625Kbps - Set for I <sup>2</sup> C Fast-mode Plus speed 0b111: 980Kbps - Set for I <sup>2</sup> C Fast-mode Plus speed
MST_TO	2:0	I <sup>2</sup> C-to-I <sup>2</sup> C main timeout setting  Internal GMSL2 I <sup>2</sup> C main times out after the configured duration if it does not receive any response while waiting for a packet from remote device.	0b000: 16µs 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled

I<sup>2</sup>C\_2 (0x42)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>			SRC_A[6:0]				–	
<b>Reset</b>			0b0000000				–	
<b>Access Type</b>			Write, Read				–	

BITFIELD	BITS	DESCRIPTION	DECODE
SRC_A	7:1	<p>I<sup>2</sup>C address translator source A for main control channel.</p> <p>When an I<sup>2</sup>C transaction across the GMSL link has a device address matching I<sup>2</sup>C SRC_A, the device address as seen on the remote side is replaced by the device address in I<sup>2</sup>C DST_A.</p>	0bXXXXXXXX: Value of I <sup>2</sup> C SRC_A

I<sup>2</sup>C\_3 (0x43)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DST_A[6:0]							
<b>Reset</b>	0b00000000							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
DST_A	7:1	<p>I<sup>2</sup>C address translator destination A for main control channel.</p> <p>See the description of I<sup>2</sup>C SRC_A.</p>			0bXXXXXXXX: Value of I <sup>2</sup> C DST_A			

I<sup>2</sup>C\_4 (0x44)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SRC_B[6:0]							
<b>Reset</b>	0b00000000							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
SRC_B	7:1	<p>I<sup>2</sup>C address translator source B for main control channel.</p> <p>When an I<sup>2</sup>C transaction across the GMSL link has a device address matching I<sup>2</sup>C SRC_B, the device address as seen on the remote side is replaced by the device address in I<sup>2</sup>C DST_B.</p>			0bXXXXXXXX: Value of I <sup>2</sup> C SRC_B			

I<sup>2</sup>C\_5 (0x45)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DST_B[6:0]							
<b>Reset</b>	0b00000000							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
DST_B	7:1	I <sup>2</sup> C address translator destination B for main control channel.  See the description of I <sup>2</sup> C SRC_B.	0bXXXXXXXX: Value of I <sup>2</sup> C DST_B

UART\_0 (0x48)\*

BIT	7	6	5	4	3	2	1	0
Field	RSVD[1:0]		REM_MS_E_N	LOC_MS_E_N	BYPASS_D_IS_PAR	BYPASS_TO[1:0]		BYPASS_E_N
Reset	0b01		0b0	0b0	0b0	0b01		0b0
Access Type			Write, Read	Write, Read	Write, Read	Write, Read		Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
REM_MS_E_N	5	Enables UART Bypass Mode Control by Remote GPIO Pin  When set, remote chip's GPIO is used as MS pin (UART mode select).  When MS is high, chip is in bypass mode, otherwise, chip is in base mode.  In bypass mode, the UART interface can only access an external UART interface through the MFP pins, and cannot access the control channel. In base mode, the control channel access is enabled.	0b0: UART bypass mode not controlled by remote MS pin 0b1: UART bypass mode controlled by remote MS pin
LOC_MS_EN	4	Enables UART bypass mode control by local GPIO pin  Set to use GPIO2 pin as MS pin (UART mode select).  When MS is high, chip is in bypass mode, otherwise, chip is in base mode.  In bypass mode, the UART interface can only access an external UART interface through the MFP pins, and cannot access the control channel. In base mode, the control channel access is enabled.	0b0: UART bypass mode not controlled by local MS pin 0b1: UART bypass mode controlled by local MS pin
BYPASS_D_IS_PAR	3	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	2:1	UART soft-bypass timeout duration	0b00: 2ms 0b01: 8ms 0b10: 32ms 0b11: Disabled (bypass mode active until next power cycle/PWDNB)

BITFIELD	BITS	DESCRIPTION	DECODE
BYPASS_EN	0	<p>Enable UART soft-bypass mode.</p> <p>Bypass mode remains active as long as there is UART activity. When there is no UART activity for selected duration (configured by BYPASS_TO bitfield), device exits bypass mode, and the bit is automatically cleared.</p>	<p>soft-bypass</p> <p>0b0: UART soft-bypass mode disabled 0b1: UART soft-bypass mode enabled</p>

I2C\_PT\_0 (0x4C)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	SLV_SH_PT[1:0]	—	—	SLV_TO_PT[2:0]	—	—
<b>Reset</b>	—	—	0b10	—	—	0b110	—	—
<b>Access Type</b>	—	—	Write, Read	—	—	Write, Read	—	—

BITFIELD	BITS	DESCRIPTION	DECODE
SLV_SH_PT	5:4	<p>Pass-through 1 and 2 I<sup>2</sup>C-to-I<sup>2</sup>C subordinate setup and hold-time setting (setup, hold).</p> <p>Configures the interval between SDA and SCL transitions when driven by the internal I<sup>2</sup>C subordinate. Set this according to the I<sup>2</sup>C speed mode.</p>	<p>0b00: Set for I<sup>2</sup>C Fast-mode Plus speed (1Mb/s) 0b01: Set for I<sup>2</sup>C Fast-mode speed (400kb/s) 0b10: Set for I<sup>2</sup>C Standard-mode speed (100kb/s) 0b11: Reserved</p>
SLV_TO_PT	2:0	<p>Pass-through 1 and 2 I<sup>2</sup>C-to-I<sup>2</sup>C subordinate timeout setting</p> <p>Internal GMSL2 I<sup>2</sup>C subordinate times out after the configured duration if it does not receive any response while waiting for a packet from remote device.</p>	<p>0b000: 16µs 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled</p>

I2C\_PT\_1 (0x4D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	—	MST_BT_PT[2:0]	—	—	MST_TO_PT[2:0]	—	—
<b>Reset</b>	0b0	—	0b101	—	—	0b110	—	—
<b>Access Type</b>	—	—	Write, Read	—	—	Write, Read	—	—

BITFIELD	BITS	DESCRIPTION	DECODE
MST_BT_PT	6:4	<p>Pass-through 1 and 2 I<sup>2</sup>C-to-I<sup>2</sup>C main bit rate setting</p> <p>Configures the I<sup>2</sup>C bit rate (SCL clock frequency) used by the internal I<sup>2</sup>C main (in the device on the remote side from the external I<sup>2</sup>C main).</p> <p>Set this according to the I<sup>2</sup>C speed mode.</p>	0b000: 9.92Kbps - Set for I <sup>2</sup> C Standard mode speed 0b001: 33.2Kbps - Set for I <sup>2</sup> C Standard mode speed 0b010: 99.2Kbps - Set for I <sup>2</sup> C Standard or Fast-mode speed 0b011: 123Kbps - Set for I <sup>2</sup> C Fast-mode speed 0b100: 203Kbps - Set for I <sup>2</sup> C Fast-mode speed 0b101: 397Kbps - Set for I <sup>2</sup> C Fast or Fast-mode Plus speed 0b110: 625Kbps - Set for I <sup>2</sup> C Fast-mode Plus speed 0b111: 980Kbps - Set for I <sup>2</sup> C Fast-mode Plus speed
MST_TO_PT	2:0	<p>Pass-through 1 and 2 I<sup>2</sup>C-to-I<sup>2</sup>C main timeout setting</p> <p>Internal GMSL2 I<sup>2</sup>C main times out after the configured duration if it does not receive any response while waiting for a packet from the remote device.</p>	0b000: 16μs 0b001: 1ms 0b010: 2ms 0b011: 4ms 0b100: 8ms 0b101: 16ms 0b110: 32ms 0b111: Disabled

#### UART\_PT\_0 (0x4F)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	BITLEN_M AN_CFG_2	DIS_PAR_2	RSVD	RSVD	BITLEN_M AN_CFG_1	DIS_PAR_1	RSVD	RSVD
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read	Write, Read			Write, Read	Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
BITLEN_MA N_CFG_2	7	Use the custom UART bit rate (selected by the BITLEN_PT_2_L - 0x54A and BITLEN_PT_2_H - 0x54B bitfields) in pass-through UART Channel 1	0b0: Use standard bit rate 0b1: Use custom bit rate
DIS_PAR_2	6	Disable parity bit in pass-through UART Channel 2	0b0: Parity bit enabled 0b1: Parity bit disabled
BITLEN_MA N_CFG_1	3	Use the custom UART bit rate (selected by the BITLEN_PT_1_L - 0x548 and BITLEN_PT_1_H - 0x549 bitfields) in pass-through UART Channel 1	0b0: Use standard bit rate 0b1: Use custom bit rate
DIS_PAR_1	2	Disable parity bit in pass-through UART Channel 1	0b0: Parity bit enabled 0b1: Parity bit disabled

[TX0 \(0x58\)\\*](#)

BIT	7	6	5	4	3	2	1	0			
<b>Field</b>	TX_CRC_EN	–	RSVD[1:0]			RSVD[1:0]					
<b>Reset</b>	0b0	–	0b11			0b00					
<b>Access Type</b>	Write, Read	–									
BITFIELD	BITS	DESCRIPTION			DECODE						
TX_CRC_EN	7	Transmit CRC Enable			0b0: Transmit CRC disabled 0b1: Transmit CRC enabled						

[TX3 \(0x5B\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	–	–	–	–	–	–	TX_STR_SEL[1:0]			
<b>Reset</b>	–	–	–	–	–	–	0b10			
<b>Access Type</b>	–	–	–	–	–	–	Write, Read			
BITFIELD	BITS	DESCRIPTION			DECODE					
TX_STR_SEL	1:0	Stream ID used in packets transmitted from this channel			0bXX: Stream ID for packets from this channel					

[TR0 \(0x78\)\\*](#)

BIT	7	6	5	4	3	2	1	0			
<b>Field</b>	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]			RSVD[1:0]					
<b>Reset</b>	0b1	0b1	0b11			0b00					
<b>Access Type</b>	Write, Read	Write, Read									
BITFIELD	BITS	DESCRIPTION			DECODE						
TX_CRC_EN	7	When set, calculate and append CRC to each packet transmitted from this port.			0b0: Transmit CRC disabled 0b1: Transmit CRC enabled						
RX_CRC_EN	6	When set, indicates that packets received at this port have appended CRC. CRC checking should be performed at each packet.			0b0: Receive CRC disabled 0b1: Receive CRC enabled						

[TR3 \(0x7B\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	–	–	–	TX_SRC_ID[2:0]		
<b>Reset</b>	–	–	–	–	–	0b000		
<b>Access Type</b>	–	–	–	–	–	Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
TX_SRC_ID	2:0	Source identifier used in packets transmitted from this channel.  Default value is based on the device address set by the CFG0 pin.	0bXXX: Source ID for packets from this channel

[TR4 \(0x7C\)\\*](#)

BIT	7	6	5	4	3	2	1	0
Field	RX_SRC_SEL[7:0]							
Reset	0xFF							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
RX_SRC_SEL	7:0	Receive packets from selected sources.  Each bit indicates whether packets with that source ID should be received or not. This is a one-hot encoding.  For example, when SRC_SEL = 00001001, then packets with source ID equal to 0 and 3 are received.	0x00: No packets received 0x01: Packets from source ID 0 received 0x02: Packets from source ID 1 received 0x03: Packets from source ID 0 and 1 received 0x04: Packets from source ID 2 received ... ... 0xFF: Packets from all source IDs received

[TR0 \(0x80, 0x90, 0xA0, 0xA8\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
Field	TX_CRC_EN	RX_CRC_EN	RSVD[1:0]			RSVD[1:0]		RSVD[1:0]		
Reset	0b1	0b1	0b11			0b00		0b00		
Access Type	Write, Read	Write, Read								

BITFIELD	BITS	DESCRIPTION	DECODE
TX_CRC_EN	7	When set, calculate and append CRC to each packet transmitted from this port.	0b0: Transmit CRC disabled 0b1: Transmit CRC enabled
RX_CRC_EN	6	When set, indicates that packets received at this port have appended CRC and CRC checking should be performed at each packet.	0b0: Receive CRC disabled 0b1: Receive CRC enabled

[TR3 \(0x83, 0x93, 0xA3, 0xAB\)\\*](#)

BIT	7	6	5	4	3	2	1	0
Field	—	—	—	—	—	TX_SRC_ID[2:0]		
Reset	—	—	—	—	—	0b000		
Access Type	—	—	—	—	—	Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
TX_SRC_ID	2:0	Source identifier used in packets transmitted from this channel.  Default value is based on the device address set by the CFG0 pin.	0bXXX: Source ID for packets from this channel

[TR4 \(0x84, 0x94, 0xA4, 0xAC\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
Field	RX_SRC_SEL[7:0]							
Reset	0xFF							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
RX_SRC_SEL	7:0	Receive packets from selected sources.  Each bit indicates whether packets with that source ID should be received or not. This is a one-hot encoding.  For example, when SRC_SEL = 00001001, then packets with source ID equal to 0 and 3 are received.	0x00: No packets received 0x01: Packets from source ID 0 received 0x02: Packets from source ID 1 received 0x03: Packets from source ID 0 and 1 received 0x04: Packets from source ID 2 received ... ... 0xFF: Packets from all source IDs received

[ARQ0 \(0x85, 0x95, 0xA5, 0xAD\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
Field	RSVD	RSVD	RSVD	RSVD	ARQ0_EN	DIS_DBL_A CK_RETX	–	–
Reset	0b1	0b0	0b0	0b1	0b1	0b0	–	–
Access Type					Write, Read	Write, Read	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
ARQ0_EN	3	Enable ARQ. It is not recommended to change this value.	0b0: ARQ disabled 0b1: ARQ enabled
DIS_DBL_A CK_RETX	2	Disable retransmission due to receiving same acknowledge twice	0b0: Enabled 0b1: Disabled

[ARQ1 \(0x86, 0x96, 0xA6, 0xAE\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
Field	–	RSVD[2:0]					MAX_RT_E RR_OEN	RT_CNT_O EN
Reset	–	0b111					0b1	0b0
Access Type	–						Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
MAX_RT_ER R_OEN	1	Enable reporting of ARQ maximum retransmission limit errors (MAX_RT_ERR) for this channel at ERRB pin.	0b0: ARQ maximum retransmission limit errors reporting at ERRB pin disabled 0b1: ARQ maximum retransmission limit errors reporting at ERRB pin enabled
RT_CNT_OE N	0	Enable reporting of ARQ retransmission event for this channel at ERRB pin.  When enabled, ERRB is asserted when RT_CNT of this channel is greater than 0.	0b0: ARQ retransmission count reporting at ERRB pin disabled 0b1: ARQ retransmission count reporting at ERRB pin enabled

**ARQ2 (0x87, 0x97, 0xA7, 0xAF)**

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	MAX_RT_E RR	RT_CNT[6:0]							
<b>Reset</b>	0b0	0b0000000							
<b>Access Type</b>	Read Clears All	Read Clears All							
BITFIELD	BITS	DESCRIPTION			DECODE				
MAX_RT_ER R	7	Reached maximum retransmission limit (MAX_RT) for one packet in this channel			0b0: Maximum retransmission limit not reached 0b1: Maximum retransmission limit reached				
RT_CNT	6:0	Total retransmission count in this channel			0xXX: Count of retransmissions for this channel				

**TR0 (0x88)\***

BIT	7	6	5	4	3	2	1	0					
<b>Field</b>	TX_CRC_E N	RX_CRC_E N	RSVD[1:0]		RSVD[1:0]		RSVD[1:0]						
<b>Reset</b>	0b1	0b1	0b11		0b00		0b00						
<b>Access Type</b>	Write, Read	Write, Read											
BITFIELD	BITS	DESCRIPTION			DECODE								
TX_CRC_EN	7	When set, calculate and append CRC to each packet transmitted from this port			0b0: Transmit CRC disabled 0b1: Transmit CRC enabled								
RX_CRC_EN	6	When set, indicates that packets received at this port have appended CRC and CRC checking should be performed at each packet.			0b0: Receive CRC disabled 0b1: Receive CRC enabled								

**TR3 (0x8B)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	TX_SRC_ID[2:0]		
<b>Reset</b>	—	—	—	—	—	0b000		
<b>Access Type</b>	—	—	—	—	—	Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
TX_SRC_ID	2:0	Source identifier used in packets transmitted from this channel.  Default value is based on the device address set by the CFG0 pin.	0bXXX: Source ID for packets from this channel

[ARQ1 \(0x8E\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	RSVD[2:0]			—	—	MAX_RT_E RR_OEN	RT_CNT_O EN
<b>Reset</b>	—	0b111			—	—	0b1	0b0
<b>Access Type</b>	—				—	—	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
MAX_RT_ER R_OEN	1	Enable reporting of ARQ max re-transmission limit errors (MAX_RT_ERR) for this channel at ERRB pin.	0b0: ARQ max re-transmit limit errors reporting at ERRB pin disabled 0b1: ARQ max re-transmit limit errors reporting at ERRB pin enabled
RT_CNT_OE N	0	Enable reporting of ARQ re-transmission event for this channel at ERRB pin. When enabled, ERRB is asserted when RT_CNT of this channel is greater than 0.	0b0: ARQ re-transmission count reporting at ERRB pin disabled 0b1: ARQ re-transmission count reporting at ERRB pin enabled

[ARQ2 \(0x8F\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MAX_RT_E RR	RT_CNT[6:0]						
<b>Reset</b>	0b0	0b0000000						
<b>Access Type</b>	Read Clears All	Read Clears All						
BITFIELD	BITS	DESCRIPTION			DECODE			
MAX_RT_ER R	7	Reached maximum re-tx limit (MAX_RT) for one packet in this channel			0b0: Max re-transmission limit not reached 0b1: Max re-transmission limit reached			
RT_CNT	6:0	Total re-transmission count in this channel			0xXX: Count of re-transmissions for this channel			

[VIDEO\\_TX0 \(0x110\)\\*](#)

BIT	7	6	5	4	3	2	1	0				
<b>Field</b>	LINE_CRC_ SEL	LINE_CRC_ EN	ENC_MODE[1:0]		AUTO_BPP	CLKDET_B YP	RSVD[1:0]					
<b>Reset</b>	0b0	0b1	0b10		0b1	0b0	0b00					
<b>Access Type</b>	Write, Read	Write, Read	Write, Read		Write, Read	Write, Read						
BITFIELD	BITS	DESCRIPTION				DECODE						
LINE_CRC_ SEL	7	Line CRC checksum generation with DE or HS				0b0: Use DE for Line CRC 0b1: Use HS for Line CRC						

BITFIELD	BITS	DESCRIPTION	DECODE
LINE_CRC_EN	6	Line CRC Enable  Generates a CRC code for the video line and sends it to the receiver side for comparison.	0b0: Line CRC disabled 0b1: Line CRC enabled
ENC_MODE	5:4	HS, VS, and DE Encoding mode. This Encoding mode is intended to minimize video bandwidth usage on the GMSL link.  When the encoding is on, the HS, VS, DE signals are included in video packets across the GMSL link only when they toggle.  Additionally, color pixel data may not be transmitted during horizontal or vertical blanking periods.  When the encoding is off, the HS, VS, DE signals are included in all video packets, along with the color pixel data, and are sent every 36 pixels.	0b00: HS, VS, DE encoding off 0b01: HS, VS, DE encoding on, color bits always sent 0b10: HS, VS, DE encoding on, color bits sent only when DE is high 0b11: HS, VS, DE encoding on, color bits sent only when HS is high
AUTO_BPP	3	Select bits per pixel (BPP) source. Set to 0 if override of BPP is required.	0b0: Use BPP from BPP register 0b1: Use BPP from MIPI receiver
CLKDET_BY_P	2	Bypass PCLK detector	

VIDEO\_TX1 (0x111)\*

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	RSVD[1:0]		BPP[5:0]						
<b>Reset</b>	0b01		0b011000						
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION					DECODE		
BPP	5:0	Color bits per pixel (RGB888 = 24)					0bXXXXXXXX: Number of bits per pixel		

VIDEO\_TX2 (0x112)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PCLKDET	DRIFT_ER_R	OVERFLO_W	FIFO_WARN	RSVD	LIM_HEART	RSVD	RSVD
<b>Reset</b>	0b0	0b0	0b0	0b0	0b1	0b0	0b1	0b0
<b>Access Type</b>	Read Only	Read Clears All	Read Clears All	Read Clears All		Write, Read		

BITFIELD	BITS	DESCRIPTION	DECODE
PCLKDET	7	<p>PCLK detected. This bit is asserted when a pixel clock can be extracted from the video interface receiver and the extracted frequency is measured to be greater than 4MHz.</p> <p>Valid in all video modes.</p> <p>Can also read the following registers to determine if MIPI data is received: phy1_pkt_cnt, csi1_pkt_cnt. In Tunneling mode, can also read tun_pkt_cnt.</p>	0b0: Video received PCLK not detected 0b1: Video received PCLK detected
DRIFT_ERR	6	<p>VID_TX PCLK drift error detected.</p> <p>After the video pipeline starts, PCLK cannot drift more than a certain amount (+/-1.25%) without restarting the subsystem.</p>	0b0: Video transmit PCLK drift error not detected 0b1: Video transmit PCLK drift error detected
OVERFLOW	5	VID_TX FIFO has overflowed, video input throughput may be too high, bandwidth allocation on GMSL link for video may not be enough. Pixel mode only.	0b0: Video transmit FIFO has not overflowed 0b1: Video transmit FIFO has overflowed
FIFO_WARN	4	VID_TX FIFO is more than half full, video data coming from the video interface receiver is read by the scheduler. Pixel mode only.	0b0: Video transmit FIFO is less than or equal to half full 0b1: Video transmit FIFO is more than half full
LIM_HEART	2	<p>Disable heartbeat during blanking</p> <p>Use together with SEQ_MISS_EN and DIS_PKT_DET bitfields in deserializer.</p>	0b0: Heartbeat enabled during blanking 0b1: Heartbeat disabled during blanking

#### SPI\_0 (0x170)\*

BIT	7	6	5	4	3	2	1	0		
Field	SPI_LOC_ID[1:0]		SPI_CC_TRG_ID[1:0]		SPI_IGNR_ID	SPI_CC_EN	MST_SLVN	SPI_EN		
Reset	0b00		0b00		0b1	0b0	0b0	0b0		
Access Type	Write, Read		Write, Read		Write, Read	Write, Read	Write, Read	Write, Read		
BITFIELD	BITS	DESCRIPTION			DECODE					
SPI_LOC_ID	7:6	Program to local ID if filtering packets based on header ID.			0b00: ID = 0 0b01: ID = 1 0b10: ID = 2 0b11: ID = 3					
SPI_CC_TRG_ID	5:4	ID for GMSL header in SPI control-channel bridge mode			0b00: ID = 0 0b01: ID = 1 0b10: ID = 2 0b11: ID = 3					
SPI_IGNR_ID	3	Selects if SPI should use or ignore header ID to decide on packet acceptance			0b0: Accept only packets with proper ID 0b1: Ignore ID and accept all packets					
SPI_CC_EN	2	Enable control channel SPI bridge function			0b0: SPI bridge disabled 0b1: SPI bridge enabled					
MST_SLVN	1	Selects if SPI is main or subordinate			0b0: SPI subordinate 0b1: SPI main					

BITFIELD	BITS	DESCRIPTION	DECODE
SPI_EN	0	Enable SPI channel	0b0: SPI channel disabled 0b1: SPI channel enabled

**SPI\_1 (0x171)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SPI_LOC_N[5:0]						SPI_BASE_PRIO[1:0]	
<b>Reset</b>	0b000111						0b01	
<b>Access Type</b>	Write, Read						Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
SPI_LOC_N	7:2	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.	0b000000: Packet size is 1 byte 0b000001: Packet size is 3 bytes ... ... 0b111111: Packet size is 127 bytes
SPI_BASE_P_RIO	1:0	Starting GMSL Request Priority, advances by 1 (if room) if Tx Buffer is over half full.	

**SPI\_2 (0x172)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REQ_HOLD_OFF[2:0]			FULL_SCK_SETUP	SPI_MOD3_F	SPI_MOD3	SPIM_SS2_ACT_H	SPIM_SS1_ACT_H
<b>Reset</b>	0b000			0b0	0b0	0b0	0b1	0b1
<b>Access Type</b>	Write, Read			Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
REQ_HOLD_OFF	7:5	The SPI port transmits data across the GMSL link as soon as it is received. This register can be used to hold off transmitting data across GMSL link until this number of extra bytes are received on the SPI port. No extra bytes means the data is transmitted as soon as it is received.	0b00: No extra bytes 0b01: 1 extra byte 0b10: 2 extra bytes 0b11: 3 extra bytes
FULL_SCK_SETUP	4	Sample MISO after half or full SCLK period.	0b0: MISO sampled after half SCLK period 0b1: MISO sampled after full SCLK period
SPI_MOD3_F	3	Allows the suppression of an extra SCLK prior to SS deassertion when SPI mode 3 is selected.	0b0: Extra SCLK present prior to SS deassertion when in SPI mode 3 0b1: Extra SCLK suppressed prior to SS deassertion when in SPI mode 3
SPI_MOD3	2	Selects SPI mode 0 or 3  In both modes, data is sampled on the rising clock edge and shifted out on the falling edge.	0b0: SPI mode 0. Clock polarity and phase are 0. Subordinate select asserts when clock is low. 0b1: SPI mode 3. Clock polarity and phase are 1. Subordinate select asserts when clock is high.
SPIM_SS2_ACT_H	1	Sets the polarity for SS2 when the SPI is a main.	0b0: Subordinate select 2 is active low 0b1: Subordinate select 2 is active high
SPIM_SS1_ACT_H	0	Sets the polarity for SS1 when the SPI is a main.	0b0: Subordinate select 1 is active low 0b1: Subordinate select 1 is active high

[SPI\\_3 \(0x173\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SPIM_SS_DLY_CLKS[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
SPIM_SS_DLY_CLKS	7:0	Number of 300MHz clock cycles to delay between: - Assertion of SS and start of SCLK pulses - End of SCLK pulses and deassertion of SS - Deassertion of SS and reassertion of SS (if necessary)			0XX: Number of clock cycles			

[SPI\\_4 \(0x174\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SPIM_SCK_LO_CLKS[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
SPIM_SCK_LO_CLKS	7:0	Number of 300MHz clock cycles for SCK low time			0XX: Number of clock cycles for SCLK low time			

[SPI\\_5 \(0x175\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SPIM_SCK_HI_CLKS[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
SPIM_SCK_HI_CLKS	7:0	Number of 300MHz clock cycles for SCLK high time			0XX: Number of clock cycles for SCLK high time			

[SPI\\_6 \(0x176\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	BNE	SPIS_RWN	SS_IO_EN_2	SS_IO_EN_1	BNE_IO_E_N	RWN_IO_E_N
<b>Reset</b>	—	—	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	—	—	Read Only	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
BNE	5	Alternate status register to use for BNE status if MFP is not available	0b0: Buffer empty 0b1: Buffer not empty
SPIS_RWN	4	Alternate GPU control register to use for Read/Write control if MFP is not available.	0b0: Write 0b1: Read
SS_IO_EN_2	3	Enable MFP for use as Subordinate Select 2 output.	0b0: MFP not used for SPI SS2 function 0b1: MFP used for SPI SS2 function
SS_IO_EN_1	2	Enable MFP for use as Subordinate Select 1 output.	0b0: MFP not used for SPI SS1 function 0b1: MFP used for SPI SS1 function
BNE_IO_EN	1	Enable MFP for use as BNE output for SPI data available status.	0b0: MFP not used for SPI BNE function 0b1: MFP used for SPI BNE function
RWN_IO_EN	0	Enable MFP for use as RO input for control of SPI data movement.	0b0: MFP not used for SPI RO function 0b1: MFP used for SPI RO function

[SPI\\_7 \(0x177\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SPI_RX_OV VRFLW	SPI_TX_OV RFLW	–	SPIS_BYTE_CNT[4:0]				
<b>Reset</b>	0b0	0b0	–	0b00000				
<b>Access Type</b>	Read Clears All	Read Clears All	–	Read Only				
BITFIELD	BITS	DESCRIPTION			DECODE			
SPI_RX_OV RFLW	7	SPI Rx Buffer Overflow Flag			0b0: No SPI Rx buffer overflow 0b1: SPI Rx buffer overflow			
SPI_TX_OV RFLW	6	SPI Tx Buffer Overflow flag			0b0: No SPI Tx buffer overflow 0b1: SPI Tx buffer overflow			
SPIS_BYTE_CNT	4:0	Number of SPI data bytes available for reading from Rx buffer.			0bXXXXX: Number of bytes available			

[SPI\\_8 \(0x178\)](#)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REQ_HOLD_OFF_TO[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
REQ_HOLD_OFF_TO	7:0	Timeout delay (in 100nS increments) for GMSL request hold off transmitting data across the GMSL link until extra bytes are received (0 is disable). See REQ_HOLD_OFF for more details.			0xXX: Number of 100nS delay increments for GMSL request hold off			

[CROSS\\_0 \(0x236\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS0_I	CROSS0_F	CROSS0[4:0]				
<b>Reset</b>	–	0b0	0b0	0b00000				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS0_I	6	Invert outgoing bit 0				0b0: Do not invert bit 0b1: Invert bit		
CROSS0_F	5	Force outgoing bit 0 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS0	4:0	Maps incoming bit position set by this field to the outgoing bit position 0				0bXXXXX: Incoming bit position		

[CROSS\\_1 \(0x237\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS1_I	CROSS1_F	CROSS1[4:0]				
<b>Reset</b>	–	0b0	0b0	0b00001				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS1_I	6	Invert outgoing bit 1				0b0: Do not invert bit 0b1: Invert bit		
CROSS1_F	5	Force outgoing bit 1 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS1	4:0	Maps incoming bit position set by this field to the outgoing bit position 1				0bXXXXX: Incoming bit position		

[CROSS\\_2 \(0x238\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS2_I	CROSS2_F	CROSS2[4:0]				
<b>Reset</b>	–	0b0	0b0	0b00010				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS2_I	6	Invert outgoing bit 2				0b0: Do not invert bit 0b1: Invert bit		
CROSS2_F	5	Force outgoing bit 2 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS2	4:0	Maps incoming bit position set by this field to the outgoing bit position 2				0bXXXXX: Incoming bit position		

[CROSS\\_3 \(0x239\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	–	CROSS3_I	CROSS3_F	CROSS3[4:0]					
<b>Reset</b>	–	0b0	0b0	0b00011					
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read					
BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS3_I	6	Invert outgoing bit 3				0b0: Do not invert bit 0b1: Invert bit			
CROSS3_F	5	Force outgoing bit 3 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS3	4:0	Maps incoming bit position set by this field to the outgoing bit position 3				0bXXXXX: Incoming bit position			

[CROSS\\_4 \(0x23A\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	–	CROSS4_I	CROSS4_F	CROSS4[4:0]					
<b>Reset</b>	–	0b0	0b0	0b00100					
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read					
BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS4_I	6	Invert outgoing bit 4				0b0: Do not invert bit 0b1: Invert bit			
CROSS4_F	5	Force outgoing bit 4 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS4	4:0	Maps incoming bit position set by this field to the outgoing bit position 4				0bXXXXX: Incoming bit position			

[CROSS\\_5 \(0x23B\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	–	CROSS5_I	CROSS5_F	CROSS5[4:0]					
<b>Reset</b>	–	0b0	0b0	0b00101					
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read					
BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS5_I	6	Invert outgoing bit 5				0b0: Do not invert bit 0b1: Invert bit			
CROSS5_F	5	Force outgoing bit 5 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS5	4:0	Maps incoming bit position set by this field to the outgoing bit position 5				0bXXXXX: Incoming bit position			

[CROSS\\_6 \(0x23C\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS6_I	CROSS6_F	CROSS6[4:0]				
<b>Reset</b>	–	0b0	0b0	0b00110				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS6_I	6	Invert outgoing bit 6				0b0: Do not invert bit 0b1: Invert bit		
CROSS6_F	5	Force outgoing bit 6 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS6	4:0	Maps incoming bit position set by this field to the outgoing bit position 6				0bXXXXX: Incoming bit position		

[CROSS\\_7 \(0x23D\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS7_I	CROSS7_F	CROSS7[4:0]				
<b>Reset</b>	–	0b0	0b0	0b00111				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS7_I	6	Invert outgoing bit 7				0b0: Do not invert bit 0b1: Invert bit		
CROSS7_F	5	Force outgoing bit 7 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS7	4:0	Maps incoming bit position set by this field to the outgoing bit position 7				0bXXXXX: Incoming bit position		

[CROSS\\_8 \(0x23E\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS8_I	CROSS8_F	CROSS8[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01000				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS8_I	6	Invert outgoing bit 8				0b0: Do not invert bit 0b1: Invert bit		
CROSS8_F	5	Force outgoing bit 8 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS8	4:0	Maps incoming bit position set by this field to the outgoing bit position 8				0bXXXXX: Incoming bit position		

[CROSS\\_9 \(0x23F\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS9_I	CROSS9_F	CROSS9[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01001				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS9_I	6	Invert outgoing bit 9				0b0: Do not invert bit 0b1: Invert bit		
CROSS9_F	5	Force outgoing bit 9 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS9	4:0	Maps incoming bit position set by this field to the outgoing bit position 9				0bXXXXX: Incoming bit position		

[CROSS\\_10 \(0x240\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS10_I	CROSS10_F	CROSS10[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01010				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS10_I	6	Invert outgoing bit 10				0b0: Do not invert bit 0b1: Invert bit		
CROSS10_F	5	Force outgoing bit 10 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS10	4:0	Maps incoming bit position set by this field to the outgoing bit position 10				0bXXXXX: Incoming bit position		

[CROSS\\_11 \(0x241\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS11_I	CROSS11_F	CROSS11[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01011				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE		
CROSS11_I	6	Invert outgoing bit 11				0b0: Do not invert bit 0b1: Invert bit		
CROSS11_F	5	Force outgoing bit 11 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero		
CROSS11	4:0	Maps incoming bit position set by this field to the outgoing bit position 11				0bXXXXX: Incoming bit position		

[CROSS\\_12 \(0x242\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS12_I	CROSS12_F	CROSS12[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01100				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS12_I	6	Invert outgoing bit 12			0b0: Do not invert bit 0b1: Invert bit			
CROSS12_F	5	Force outgoing bit 12 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS12	4:0	Maps incoming bit position set by this field to the outgoing bit position 12			0bXXXXX: Incoming bit position			

[CROSS\\_13 \(0x243\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS13_I	CROSS13_F	CROSS13[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01101				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS13_I	6	Invert outgoing bit 13			0b0: Do not invert bit 0b1: Invert bit			
CROSS13_F	5	Force outgoing bit 13 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS13	4:0	Maps incoming bit position set by this field to the outgoing bit position 13			0bXXXXX: Incoming bit position			

[CROSS\\_14 \(0x244\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS14_I	CROSS14_F	CROSS14[4:0]				
<b>Reset</b>	–	0b0	0b0	0b01110				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS14_I	6	Invert outgoing bit 14			0b0: Do not invert bit 0b1: Invert bit			
CROSS14_F	5	Force outgoing bit 14 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			

BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS14	4:0	Maps incoming bit position set by this field to the outgoing bit position 14				0bXXXXX: Incoming bit position			

[CROSS\\_15 \(0x245\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS15_I	CROSS15_F	CROSS15[4:0]				
<b>Reset</b>	—	0b0	0b0	0b01111				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS15_I	6	Invert outgoing bit 15				0b0: Do not invert bit 0b1: Invert bit			
CROSS15_F	5	Force outgoing bit 15 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS15	4:0	Maps incoming bit position set by this field to the outgoing bit position 15				0bXXXXX: Incoming bit position			

[CROSS\\_16 \(0x246\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS16_I	CROSS16_F	CROSS16[4:0]				
<b>Reset</b>	—	0b0	0b0	0b10000				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS16_I	6	Invert outgoing bit 16				0b0: Do not invert bit 0b1: Invert bit			
CROSS16_F	5	Force outgoing bit 16 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.				0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS16	4:0	Maps incoming bit position set by this field to the outgoing bit position 16				0bXXXXX: Incoming bit position			

[CROSS\\_17 \(0x247\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS17_I	CROSS17_F	CROSS17[4:0]				
<b>Reset</b>	—	0b0	0b0	0b10001				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION				DECODE			
CROSS17_I	6	Invert outgoing bit 17				0b0: Do not invert bit 0b1: Invert bit			

BITFIELD	BITS	DESCRIPTION	DECODE
CROSS17_F	5	Force outgoing bit 17 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.	0b0: Do not force bit to zero 0b1: Force bit to zero
CROSS17	4:0	Maps incoming bit position set by this field to the outgoing bit position 17	0bXXXXX: Incoming bit position

[CROSS\\_18 \(0x248\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS18_I	CROSS18_F	CROSS18[4:0]				
<b>Reset</b>	—	0b0	0b0	0b10010				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS18_I	6	Invert outgoing bit 18			0b0: Do not invert bit 0b1: Invert bit			
CROSS18_F	5	Force outgoing bit 18 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS18	4:0	Maps incoming bit position set by this field to the outgoing bit position 18			0bXXXXX: Incoming bit position			

[CROSS\\_19 \(0x249\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS19_I	CROSS19_F	CROSS19[4:0]				
<b>Reset</b>	—	0b0	0b0	0b10011				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS19_I	6	Invert outgoing bit 19			0b0: Do not invert bit 0b1: Invert bit			
CROSS19_F	5	Force outgoing bit 19 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS19	4:0	Maps incoming bit position set by this field to the outgoing bit position 19			0bXXXXX: Incoming bit position			

[CROSS\\_20 \(0x24A\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS20_I	CROSS20_F	CROSS20[4:0]				
<b>Reset</b>	—	0b0	0b0	0b10100				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
CROSS20_I	6	Invert outgoing bit 20	0b0: Do not invert bit 0b1: Invert bit
CROSS20_F	5	Force outgoing bit 20 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.	0b0: Do not force bit to zero 0b1: Force bit to zero
CROSS20	4:0	Maps incoming bit position set by this field to the outgoing bit position 20	0bXXXXX: Incoming bit position

**CROSS 21 (0x24B)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS21_I	CROSS21_F	CROSS21[4:0]				
<b>Reset</b>	–	0b0	0b0	0b10101				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS21_I	6	Invert outgoing bit 21			0b0: Do not invert bit 0b1: Invert bit			
CROSS21_F	5	Force outgoing bit 21 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS21	4:0	Maps incoming bit position set by this field to the outgoing bit position 21			0bXXXXX: Incoming bit position			

**CROSS 22 (0x24C)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSS22_I	CROSS22_F	CROSS22[4:0]				
<b>Reset</b>	–	0b0	0b0	0b10110				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSS22_I	6	Invert outgoing bit 22			0b0: Do not invert bit 0b1: Invert bit			
CROSS22_F	5	Force outgoing bit 22 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSS22	4:0	Maps incoming bit position set by this field to the outgoing bit position 22			0bXXXXX: Incoming bit position			

[CROSS\\_23 \(0x24D\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	CROSS23_I	CROSS23_F	CROSS23[4:0]				
<b>Reset</b>	—	0b0	0b0	0b10111				
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
CROSS23_I	6	Invert outgoing bit 23	0b0: Do not invert bit 0b1: Invert bit
CROSS23_F	5	Force outgoing bit 23 to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.	0b0: Do not force bit to zero 0b1: Force bit to zero
CROSS23	4:0	Maps incoming bit position set by this field to the outgoing bit position 23	0bXXXXX: Incoming bit position

[VTX0 \(0x24E\)\\*](#)

Used to generate sync signals for the image sensor. Uses reference clock output (derived from XTAL clock) as reference to generate sync signals for the image sensor.

REF\_VTG works the same as regular VTG, with the exception of how they are connected internally. VTG uses the input pixel clock as the clock source, and HS, VS, DE signals generated can replace the HS, VS, DE signals received in the input video.

REF\_VTG uses the REFGEN\_PLL output as the clock source. REFGEN\_PLL is a DPLL that uses XTAL clock as the clock source. It can be programmed to generate any PCLK frequency. PCLK generated by REFGEN\_PLL (RCLK in pin description) and HS, VS, DE generated by REF\_VTG are synchronous to RCLK and output from MFP pins, when enabled. These are then used by the image sensor, which drives the CSI-2 input of the serializer.

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	GEN_VS	GEN_HS	GEN_DE	VS_INV	HS_INV	DE_INV	VTG_MODE[1:0]	
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b11	
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
GEN_VS	7	Enable to generate VS output according to the timing definition	0b0: Do not generate VS 0b1: Generate VS
GEN_HS	6	Enable to generate HS output according to the timing definition	0b0: Do not generate HS 0b1: Generate HS
GEN_DE	5	Enable to generate DE output according to the timing definition	0b0: Do not generate DE 0b1: Generate DE
VS_INV	4	Invert VSYNC output of video-timing generator	0b0: Do not invert VS 0b1: Invert VS
HS_INV	3	Invert HSYNC output of video-timing generator	0b0: Do not invert HS 0b1: Invert HS
DE_INV	2	Invert DE output of video-timing generator	0b0: Do not invert DE 0b1: Invert DE

BITFIELD	BITS	DESCRIPTION	DECODE
VTG_MODE	1:0	<p>Video interface timing-generation mode. Used when VTG GEN_VS, GEN_HS, or GEN_DE are enabled.</p> <p>00 = VS tracking mode. VS input's period (VS_HIGH + VS_LOW) is tracked. After VS tracking is locked, any VS input edge (glitches) not in the expected PCLK cycle is ignored. VS tracking is locked with three consecutive matches and unlocked by three consecutive mismatches. When unlocked or powered up, the next VS input edge is assumed to be the right VS edge.</p> <p>01 = VS one-trigger mode (default) One VS input edge triggers the generation of one frame of VS/HS/DE output. If the next VS input edge comes earlier or later than expected by VS period, the newly generated frame is correct. The current VS/HS/DE output is cut or extended at the time point of the rising edge of the newly generated VS/HS/DE output.</p> <p>10 = Auto-repeat mode VS input edge triggers the generation of continuous frames of VS/HS/DE output even if no more VS input edges are seen. If the next VS input edge comes earlier or later than expected by VS period, the newly generated frame is correct. The current VS/HS/DE output is cut or extended at the time point of the rising edge of the newly generated VS/HS/DE output.</p>	<p>0b00: VS tracking mode 0b01: VS one trigger mode 0b10: Auto-repeat mode 0b11: Free running mode</p>

### VTX1 (0x24F)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	PCLKDET_VTX	–	PATGEN_CLK_SRC[2:0]			VS_TRIG
<b>Reset</b>	–	–	0b0	–	0b000			0b1
<b>Access Type</b>	–	–	Read Only	–	Write, Read			Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
PCLKDET_VTX	5	<p>PCLK detected. This bit is asserted when a pixel clock can be extracted from the video interface receiver, and the extracted frequency is measured to be greater than 4MHz.</p> <p>Valid in all video modes.</p> <p>Also read the following registers to determine if MIPI data is received: phy1_pkt_cnt, csi1_pkt_cnt.</p> <p>In Tunneling mode, tun_pkt_cnt can be read.</p>	<p>0b0: PCLK not detected 0b1: PCLK detected</p>

BITFIELD	BITS	DESCRIPTION	DECODE
PATGEN_CLK_SRC	3:1	Pattern generator clock source for video PRBS, checkerboard, and gradient patterns.	3'b0XX: Use external clock 3'b100: Use 25MHz internal clock 3'b101: Use 75MHz internal clock 3'b110: Use 150MHz internal clock 3'b111: Use 375MHz internal clock
VS_TRIG	0	Select VS trigger edge (positive vs. negative polarity of VS)	0b0: Falling edge 0b1: Rising edge

VTX2 (0x250)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_DLY_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
VS_DLY_2	7:0	VS delay in terms of PCLK cycles The output VS is delayed by VS_DLY cycles from the input VS. (bits [23:16])						0XX: Most significant byte of VS_DLY

VTX3 (0x251)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_DLY_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
VS_DLY_1	7:0	VS delay in terms of PCLK cycles The output VS is delayed by VS_DLY cycles from the input VS. (bits [15:8])						0XX: Middle significant byte of VS_DLY

VTX4 (0x252)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_DLY_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
VS_DLY_0	7:0	VS delay in terms of PCLK cycles The output VS is delayed by VS_DLY cycles from the input VS. (bits [7:0])						0XX: Least significant byte of VS_DLY

VTX5 (0x253)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_HIGH_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
VS_HIGH_2	7:0	VS high period in terms of PCLK cycles (bits [23:16])				0xXX: Most significant byte of VS high period		

VTX6 (0x254)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_HIGH_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
VS_HIGH_1	7:0	VS high period in terms of PCLK cycles (bits [15:8])				0xXX: Middle significant byte of VS high period		

VTX7 (0x255)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_HIGH_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
VS_HIGH_0	7:0	VS high period in terms of PCLK cycles (bits [7:0])				0xXX: Least significant byte of VS high period		

VTX8 (0x256)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_LOW_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
VS_LOW_2	7:0	VS low period in terms of PCLK cycles (bits [23:16])				0xXX: Most significant byte of VS low period		

VTX9 (0x257)\*

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	VS_LOW_1[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
VS_LOW_1	7:0	VS low period in terms of PCLK cycles (bits [15:8])					0xXX: Middle significant byte of VS low period		

VTX10 (0x258)\*

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	VS_LOW_0[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
VS_LOW_0	7:0	VS low period in terms of PCLK cycles (bits [7:0])					0xXX: Least significant byte of VS low period		

VTX11 (0x259)\*

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	V2H_2[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
V2H_2	7:0	VS edge to the rising edge of the first HS in terms of PCLK cycles (bits [23:16])					0xXX: Most significant byte of VS edge to first HS rising edge		

VTX12 (0x25A)\*

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	V2H_1[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
V2H_1	7:0	VS edge to the rising edge of the first HS in terms of PCLK cycles (bits [15:8])					0xXX: Middle significant byte of VS edge to first HS rising edge		

VTX13 (0x25B)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	V2H_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
V2H_0	7:0	VS edge to the rising edge of the first HS in terms of PCLK cycles (bits [7:0])			0xXX: Least significant byte of VS edge to first HS rising edge			

VTX14 (0x25C)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_HIGH_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_HIGH_1	7:0	HS high period in terms of PCLK cycles (bits [15:8])			0xXX: Most significant byte of HS high period			

VTX15 (0x25D)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_HIGH_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_HIGH_0	7:0	HS high period in terms of PCLK cycles (bits [7:0])			0xXX: Least significant byte of HS high period			

VTX16 (0x25E)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_LOW_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_LOW_1	7:0	HS low period in terms of PCLK cycles (bits [15:8])			0xXX: Most significant byte of HS low period			

[VTX17 \(0x25F\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_LOW_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
HS_LOW_0	7:0	HS low period in terms of PCLK cycles (bits [7:0])				0xXX: Least significant byte of HS low period		

[VTX18 \(0x260\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_CNT_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
HS_CNT_1	7:0	HS pulses per frame (bits [15:8])				0xXX: Most significant byte of HS pulses per frame		

[VTX19 \(0x261\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_CNT_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
HS_CNT_0	7:0	HS pulses per frame (bits [7:0])				0xXX: Least significant byte of HS pulses per frame		

[VTX20 \(0x262\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	V2D_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
V2D_2	7:0	VS edge to the rising edge of the first DE in terms of PCLK cycles (bits [23:16])				0xXX: Most significant byte of VS edge to first DE		

VTX21 (0x263)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	V2D_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
V2D_1	7:0	VS edge to the rising edge of the first DE in terms of PCLK cycles (bits [15:8])			0xXX: Middle significant byte of VS edge to first DE			

VTX22 (0x264)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	V2D_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
V2D_0	7:0	VS edge to the rising edge of the first DE in terms of PCLK cycles (bits [7:0])			0xXX: Least significant byte of VS edge to first DE			

VTX23 (0x265)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DE_HIGH_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
DE_HIGH_1	7:0	DE high period in terms of PCLK cycles (bits [15:8])			0xXX: Most significant byte of DE high period			

VTX24 (0x266)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DE_HIGH_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
DE_HIGH_0	7:0	DE high period in terms of PCLK cycles (bits [7:0])			0xXX: Least significant byte of DE high period			

**VTX25 (0x267)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DE_LOW_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
DE_LOW_1	7:0	DE low period in terms of PCLK cycles (bits [15:8])				0xXX: Most significant byte of DE low period		

**VTX26 (0x268)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DE_LOW_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
DE_LOW_0	7:0	DE low period in terms of PCLK cycles (bits [7:0])				0xXX: Least significant byte of DE low period		

**VTX27 (0x269)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DE_CNT_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
DE_CNT_1	7:0	Active lines per frame (DE pulses) (bits [15:8])				0xXX: Most significant byte of DE pulses per frame		

**VTX28 (0x26A)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DE_CNT_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>				<b>DECODE</b>		
DE_CNT_0	7:0	Active lines per frame (DE pulses) (bits [7:0])				0xXX: Least significant byte of DE pulses per frame		

VTX29 (0x26B)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VID_PRBS_EN	RSVD	VPRBS_FAI_L	–	–	GRAD_MODE	PATGEN_MODE[1:0]	
<b>Reset</b>	0b0	0b0	0b0	–	–	0b0	0b00	
<b>Access Type</b>	Write, Read		Read Only	–	–	Write, Read	Write, Read	
BITFIELD	BITS	DESCRIPTION					DECODE	
VID_PRBS_EN	7	Enable video PRBS generator					0b0: Video PRBS generator disabled 0b1: Video PRBS generator enabled	
VPRBS_FAIL	5	Video PRBS check pass/fail					0b0: Video PRBS check passed 0b1: Video PRBS check failed	
GRAD_MODE	2	Gradient pattern-generator mode					0b0: Gradient mode increasing. Each gradient color starts from a value of 0x00 and increases to 0xFF 0b1: Gradient mode decreasing. Each gradient color starts from a value of 0xFF and decreases to 0x00	
PATGEN_MODE	1:0	Pattern-generator mode					0b00: Pattern generator disabled - use video from the serializer input 0b01: Generate checkerboard pattern 0b10: Generate gradient pattern 0b11: Reserved	

VTX30 (0x26C)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	GRAD_INC[7:0]							
<b>Reset</b>	0x04							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION					DECODE	
GRAD_INC	7:0	Gradient mode increment amount (increment amount is the register value divided by 4)					0XX: Gradient increment base	

VTX31 (0x26D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	CHKR_A_L[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION					DECODE	
CHKR_A_L	7:0	Checkerboard Mode Color A Low Byte					0XX: Least significant byte of checkerboard mode color A	

VTX32 (0x26E)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_A_M[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
CHKR_A_M	7:0	Checkerboard Mode Color A Middle Byte					0xXX: Middle significant byte of checkerboard mode color A		

VTX33 (0x26F)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_A_H[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
CHKR_A_H	7:0	Checkerboard Mode Color A High Byte					0xXX: Most significant byte of checkerboard mode color A		

VTX34 (0x270)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_B_L[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
CHKR_B_L	7:0	Checkerboard Mode Color B Low Byte					0xXX: Least significant byte of checkerboard mode color B		

VTX35 (0x271)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_B_M[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
CHKR_B_M	7:0	Checkerboard Mode Color B Middle Byte					0xXX: Middle significant byte of checkerboard mode color B		

[VTX36 \(0x272\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_B_H[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION							DECODE
CHKR_B_H	7:0	Checkerboard Mode Color B High Byte							0xXX: Most significant byte of checkerboard mode color B

[VTX37 \(0x273\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_RPT_A[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION							DECODE
CHKR_RPT_A	7:0	Checkerboard Mode Color A: Dimension of each square in number of pixels							0xXX: Repeat count of checkerboard mode color A

[VTX38 \(0x274\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_RPT_B[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION							DECODE
CHKR_RPT_B	7:0	Checkerboard Mode Color B: Dimension of each square in number of pixels Set equal to CHKR_RPT_A for square checkerboard pattern							0xXX: Repeat count of checkerboard mode color B

[VTX39 \(0x275\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	CHKR_ALT[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION							DECODE
CHKR_ALT	7:0	Checkerboard Mode Alternate Line Count: Dimension of each square in number of video lines Set equal to CHKR_RPT_A for square checkerboard pattern							0xXX: Checkerboard mode alternate line count

VTX40 (0x276)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	CROSSHS_I	CROSSHS_F	CROSSHS[4:0]				
<b>Reset</b>	0b0	0b0	0b0	0b11000				
<b>Access Type</b>		Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSSHS_I	6	Invert outgoing HS			0b0: Do not invert bit 0b1: Invert bit			
CROSSHS_F	5	Force HS to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.						
CROSSHS	4:0	Map selected internal signal to HS			0bXXXXX: Incoming bit position			

VTX41 (0x277)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSSVS_I	CROSSVS_F	CROSSVS[4:0]				
<b>Reset</b>	–	0b0	0b0	0b11001				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSSVS_I	6	Invert outgoing VS			0b0: Do not invert bit 0b1: Invert bit			
CROSSVS_F	5	Force VS to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSSVS	4:0	Map selected internal signal to VS			0bXXXXX: Incoming bit position			

VTX42 (0x278)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	CROSSDE_I	CROSSDE_F	CROSSDE[4:0]				
<b>Reset</b>	–	0b0	0b0	0b11010				
<b>Access Type</b>	–	Write, Read	Write, Read	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
CROSSDE_I	6	Invert outgoing DE			0b0: Do not invert bit 0b1: Invert bit			
CROSSDE_F	5	Force DE to 0. Applied before inversion so that if inversion is also set, the outgoing bit is forced to 1.			0b0: Do not force bit to zero 0b1: Force bit to zero			
CROSSDE	4:0	Map selected internal signal to DE			0bXXXXX: Incoming bit position			

[GPIO\\_A \(0x2BE\)\\*](#)

GPIO 0

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b1	0b1	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pullup/pulldown strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_E_N	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_E_N	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_E_N	1	GPIO Tx source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

[GPIO\\_B \(0x2BF\)\\*](#)

GPIO 0

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PULL_UPDN_SEL[1:0]	OUT_TYPE			GPIO_TX_ID[4:0]			
<b>Reset</b>	0b10	0b1			0b00000			
<b>Access Type</b>	Write, Read	Write, Read			Write, Read			

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pullup/pulldown configuration	0b00: None 0b01: Pullup 0b10: Pulldown 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

[GPIO\\_C \(0x2C0\)\\*](#)

GPIO 0

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECVED	–	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	–	0b00000				
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECVED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2C1)\*

#### GPIO 1

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b0	0b0	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor Pullup/Pulldown Strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_E_N	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_E_N	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_E_N	1	GPIO Tx source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2C2)\*

#### GPIO 1

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]				
<b>Reset</b>	0b00		0b1	0b00001				
<b>Access Type</b>	Write, Read		Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pullup/pulldown configuration	0b00: None 0b01: Pullup 0b10: Pulldown 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2C3)\*

#### GPIO 1

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECV_ED	GPIO_RECV_ED	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	–	0b00001				
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting.  When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECV_ED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2C4)\*

#### GPIO 2

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b1	0b1	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pullup/pulldown strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_EN	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_EN	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_EN	1	GPIO Tx source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

[GPIO\\_B \(0x2C5\)\\*](#)

## GPIO 2

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]					
<b>Reset</b>	0b00		0b1	0b00010					
<b>Access Type</b>	Write, Read		Write, Read	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pullup/pulldown configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

[GPIO\\_C \(0x2C6\)\\*](#)

## GPIO 2

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	OVR_RES_CFG	GPIO_REC_VED	–	GPIO_RX_ID[4:0]					
<b>Reset</b>	0b0	0b1	–	0b00010					
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_RECV_ED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2C7)\*

GPIO 3

BIT	7	6	5	4	3	2	1	0
Field	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
Reset	0b1	0b0	0b0	0b0	0b0	0b0	0b0	0b1
Access Type	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pullup/pulldown strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_E_N	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_E_N	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_E_N	1	GPIO Tx source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2C8)\*

GPIO 3

BIT	7	6	5	4	3	2	1	0	
Field	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]					
Reset	0b10		0b1	0b00011					
Access Type	Write, Read		Write, Read	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pullup/pulldown configuration	0b00: None 0b01: Pullup 0b10: Pulldown 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2C9)\*

## GPIO 3

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECVED	–	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	–	0b00011				
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting.  When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECVED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2CA)\*

## GPIO 4

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b1	0b1	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pullup/pulldown strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_E_N	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_E_N	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_E_N	1	GPIO Tx source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2CB)\*

GPIO 4

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]				
<b>Reset</b>	0b10		0b1	0b00100				
<b>Access Type</b>	Write, Read		Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2CC)\*

GPIO 4

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG		GPIO_RECV_ED	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0		0b1	0b00100				
<b>Access Type</b>	Write, Read		Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_C FG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECV_ED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	

### GPIO\_A (0x2CD)\*

GPIO 5

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b0	0b0	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pull-up/pull-down strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_EN	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_EN	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_EN	1	GPIO TX source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2CE)\*

## GPIO 5

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]					
<b>Reset</b>	0b10		0b1	0b00101					
<b>Access Type</b>	Write, Read		Write, Read	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2CF)\*

## GPIO 5

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECVED	–	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	–	0b00101				
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECVED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2D0)\*

GPIO 6

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b1	0b1	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pull-up/pull-down strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_E_N	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_E_N	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_E_N	1	GPIO TX source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2D1)\*

GPIO 6

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]				
<b>Reset</b>	0b10		0b1	0b00110				
<b>Access Type</b>	Write, Read		Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2D2)\*

#### GPIO 6

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECV_ED	GPIO_RECV_ED	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	—	0b00110				
<b>Access Type</b>	Write, Read	Write, Read	—	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting.  When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECV_ED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2D3)\*

#### GPIO 7

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b0	0b0	0b0	0b1	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pull-up/pull-down strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_EN	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_EN	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_EN	1	GPIO TX source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

[GPIO\\_B \(0x2D4\)\\*](#)

## GPIO 7

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]					
<b>Reset</b>	0b10		0b1	0b00111					
<b>Access Type</b>	Write, Read		Write, Read	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

[GPIO\\_C \(0x2D5\)\\*](#)

## GPIO 7

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	OVR_RES_CFG	GPIO_REC_VED	–	GPIO_RX_ID[4:0]					
<b>Reset</b>	0b0	0b1	–	0b00111					
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_RECV_ED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2D6)\*

GPIO 8

BIT	7	6	5	4	3	2	1	0
Field	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
Reset	0b1	0b0	0b0	0b1	0b1	0b1	0b0	0b0
Access Type	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pull-up/pull-down strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_E_N	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_E_N	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_E_N	1	GPIO TX source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2D7)\*

GPIO 8

BIT	7	6	5	4	3	2	1	0	
Field	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]					
Reset	0b00		0b1	0b01000					
Access Type	Write, Read		Write, Read	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2D8)\*

## GPIO 8

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECVED	–	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	–	0b01000				
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting.  When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECVED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### GPIO\_A (0x2D9)\*

## GPIO 9

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b0	0b0	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pull-up/pull-down strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_EN	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_EN	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_EN	1	GPIO TX source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission

BITFIELD	BITS	DESCRIPTION	DECODE
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### [GPIO\\_B \(0x2DA\)\\*](#)

GPIO 9

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]				
<b>Reset</b>	0b10		0b1	0b01001				
<b>Access Type</b>	Write, Read		Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### [GPIO\\_C \(0x2DB\)\\*](#)

GPIO 9

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG		GPIO_RECV_ED	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0		0b1	0b01001				
<b>Access Type</b>	Write, Read		Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_C FG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECV_ED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### [GPIO\\_A \(0x2DC\)\\*](#)

GPIO 10

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RES_CFG	RSVD	TX_COMP_EN	GPIO_OUT	GPIO_IN	GPIO_RX_EN	GPIO_TX_EN	GPIO_OUT_DIS
<b>Reset</b>	0b1	0b0	0b0	0b1	0b1	0b0	0b0	0b1
<b>Access Type</b>	Write, Read		Write, Read	Write, Read	Read Only	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RES_CFG	7	Resistor pull-up/pull-down strength	0b0: 40kΩ 0b1: 1MΩ
TX_COMP_EN	5	Jitter minimization compensation enable	0b0: Jitter compensation disabled 0b1: Jitter compensation enabled
GPIO_OUT	4	GPIO pin output drive value when GPIO_RX_EN = 0. This can be used to drive a value out on an MFP.	0b0: This GPIO pin output is driven to 0 0b1: This GPIO pin output is driven to 1
GPIO_IN	3	GPIO pin local MFP input level	0b0: This GPIO pin value is 0 0b1: This GPIO pin value is 1
GPIO_RX_EN	2	GPIO out source control. Set to 1 to receive GPIO value from GMSL link. Set GPIO_OUT_DIS to 0 to output the received value on the local MFP.	0b0: This GPIO source disabled for GMSL reception 0b1: This GPIO source enabled for GMSL reception
GPIO_TX_EN	1	GPIO TX source control	0b0: This GPIO source disabled for GMSL transmission 0b1: This GPIO source enabled for GMSL transmission
GPIO_OUT_DIS	0	Disable GPIO output driver	0b0: Output driver enabled 0b1: Output driver disabled

### GPIO\_B (0x2DD)\*

GPIO 10

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	PULL_UPDN_SEL[1:0]		OUT_TYPE	GPIO_TX_ID[4:0]					
<b>Reset</b>	0b00		0b1	0b01010					
<b>Access Type</b>	Write, Read		Write, Read	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
PULL_UPDN_SEL	7:6	Buffer pull up/down configuration	0b00: None 0b01: Pull-up 0b10: Pull-down 0b11: Reserved
OUT_TYPE	5	Driver type selection	0b0: Open-drain 0b1: Push-pull
GPIO_TX_ID	4:0	GPIO ID for pin while transmitting	0bXXXXX: This GPIO transmit ID

### GPIO\_C (0x2DE)\*

GPIO 10

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OVR_RES_CFG	GPIO_RECVED	–	GPIO_RX_ID[4:0]				
<b>Reset</b>	0b0	0b1	–	0b01010				
<b>Access Type</b>	Write, Read	Write, Read	–	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
OVR_RES_CFG	7	Override non-GPIO port function IO setting. When set, RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS are effective when pin is configured as non-GPIO. When cleared, non-GPIO pin function determines IO type.	0b0: Non-GPIO function determines IO type when alternate function is selected 0b1: RES_CFG, PULL_UPDN_SEL, OUT_TYPE, and GPIO_OUT_DIS determine IO type for non-GPIO configuration
GPIO_RECVED	6	Received GPIO value from across the GMSL link	
GPIO_RX_ID	4:0	GPIO ID for pin while receiving	0bXXXXX: This GPIO receive ID

### CMU2 (0x302)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	PFDDIV_RSHORT[2:0]					RSVD[1:0]	RSVD
<b>Reset</b>	0b0	0b000					0b00	0b0
<b>Access Type</b>		Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
PFDDIV_RS_HORT	6:4	PFDDIV regulator voltage control.	0: VREG_PFDDIV = 1.0V 1: VREG_PFDDIV = 1.1V 2: VREG_PFDDIV = 0.875V 3: VREG_PFDDIV = 0.94V 4: VREG_PFDDIV = 1.0V 5: VREG_PFDDIV = 1.1V 6: VREG_PFDDIV = 0.875V 7: VREG_PFDDIV = 0.94V

### FRONTTOP\_0 (0x308)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	enable_line_info	START_POR_TB	–	–	RSVD	–	–
<b>Reset</b>	0b0	0b1	0b1	–	–	0b1	–	–
<b>Access Type</b>		Write, Read	Write, Read	–	–		–	–
BITFIELD	BITS	DESCRIPTION					DECODE	
enable_line_info	6	Enable sending line start info-frames					0b0: Line start info frames disabled 0x1: Line start info frames enabled	
START_POR_TB	5	Enable CSI Port					0b0: CSI disabled 0x1: CSI enabled	

FRONTTOP\_5 (0x30D)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VC_SELZ_L[7:0]							
<b>Reset</b>	0xFF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS			DESCRIPTION				
VC_SELZ_L	7:0			Virtual channel filter bits [7:0]. Each bit represents whether a virtual channel's packets are processed or discarded (bit 0 is virtual Channel 0, bit 1 is virtual Channel 1, etc.). If the bit is set to 1, the virtual channel packets are processed. If the bit is set to 0, the virtual channel packets are discarded.				

FRONTTOP\_6 (0x30E)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VC_SELZ_H[7:0]							
<b>Reset</b>	0xFF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS			DESCRIPTION			DECODE	
VC_SELZ_H	7:0			Virtual channel filter bits [15:8]. Each bit represents whether a virtual channel's packets are processed or discarded (bit 0 is virtual Channel 0, bit 1 is virtual Channel 1 etc.). If the bit is set to 1, the virtual channel packets are processed. If the bit is set to 0, the virtual channel packets are discarded			0bXXXX_XXX0: Virtual Channel 8 video packets are discarded 0bXXXX_XXX1: Virtual Channel 8 video packets are processed 0bXXXX_XX0X: Virtual Channel 9 video packets are discarded 0bXXXX_XX1X: Virtual Channel 9 video packets are processed 0bXXXX_X0XX: Virtual Channel 10 video packets are discarded 0bXXXX_X1XX: Virtual Channel 10 video packets are processed 0bXXXX_0XXX: Virtual Channel 11 video packets are discarded 0bXXXX_1XXX: Virtual Channel 11 video packets are processed 0bXX0_XXXX: Virtual Channel 12 video packets are discarded 0bXX1_XXXX: Virtual Channel 12 video packets are processed 0bXX0X_XXXX: Virtual Channel 13 video packets are discarded 0bXX1X_XXXX: Virtual Channel 13 video packets are processed 0bX0XX_XXXX: Virtual Channel 14 video packets are discarded 0bX1XX_XXXX: Virtual Channel 14 video packets are processed 0b0XXX_XXXX: Virtual Channel 15 video packets are discarded 0b1XXX_XXXX: Virtual Channel 15 video packets are processed	

FRONTTOP\_9 (0x311)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	START PO RTBZ	–	–	–	–	–	–
<b>Reset</b>	–	0b1	–	–	–	–	–	–
<b>Access Type</b>	–	Write, Read	–	–	–	–	–	–
BITFIELD	BITS	DESCRIPTION						DECODE
START_POR TBZ	6	Start video pipe Z from CSI port						0b0: Video not started 0b1: Start video

FRONTTOP\_10 (0x312)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	RSVD	–	–	–	bpp8dblz	–	–
<b>Reset</b>	–	0b0	–	–	–	0b0	–	–
<b>Access Type</b>	–	–	–	–	–	Write, Read	–	–
BITFIELD	BITS	DESCRIPTION						DECODE
bpp8dblz	2	Send 8-bit pixels as 16-bit on video pipe Z (double pixel mode). Maximizes GMSL link bandwidth capacity for 8-bit pixels.						0b0: Send as 8-bit pixels 0b1: Send 8-bit pixels as 16-bit

FRONTTOP\_11 (0x313)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	bpp12dblz	–	–	–	bpp10dblz	–	–
<b>Reset</b>	–	0b0	–	–	–	0b0	–	–
<b>Access Type</b>	–	Write, Read	–	–	–	Write, Read	–	–
BITFIELD	BITS	DESCRIPTION						DECODE
bpp12dblz	6	Send 12-bit pixels as 24-bit on video pipe Z (double pixel mode). Maximizes GMSL link bandwidth capacity for 12-bit pixels.						0b0: Send as 12-bit pixels 0b1: Send 12-bit pixels as 24-bit
bpp10dblz	2	Send 10-bit pixels as 20-bit on video pipe Z (double pixel mode). Maximizes GMSL link bandwidth capacity for 10-bit pixels.						0b0: Send as 10-bit pixels 0b1: Send 10-bit pixels as 20-bit

FRONTTOP\_16 (0x318)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	mem_dt1_selz[6:0]						–
<b>Reset</b>	–	0b0000000						–
<b>Access Type</b>	–	Write, Read						–

BITFIELD	BITS	DESCRIPTION	DECODE
mem_dt1_selz	6:0	Select designated datatype to route to video pipeline Z. Bit 6 is the enable. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt1_selz[6], mem_dt2_selz[6], mem_dt7_selz[6], and mem_dt8_selz[6] are all 0.	0b0XXXXXX: Datatype selection disabled 0b1XXXXXX: Datatype enabled for datatype selected to route to video pipeline

FRONTTOP\_17 (0x319)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	mem_dt2_selz[6:0]						
<b>Reset</b>	—	0b0000000						
<b>Access Type</b>	—	Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
mem_dt2_selz	6:0	Select designated datatype to route to video pipeline Z. Bit 6 is the enable. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt1_selz[6], mem_dt2_selz[6], mem_dt7_selz[6], and mem_dt8_selz[6] are all 0.	0b0XXXXXX: Datatype selection disabled 0b1XXXXXX: Datatype enabled for datatype selected to route to video pipeline

FRONTTOP\_22 (0x31E)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	soft_dtz_en	soft_vcZ_en	soft_bppz_en	soft_bppz[4:0]				
<b>Reset</b>	0b0	0b0	0b0	0b11000				
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read				

BITFIELD	BITS	DESCRIPTION	DECODE
soft_dtz_en	7	Datatype software override enable for video pipeline Z	0b0: Software override disabled 0b1: Software override enabled
soft_vcZ_en	6	Virtual channel software override enable for video pipeline Z	0b0: Software override disabled 0b1: Software override enabled
soft_bppz_en	5	BPP software override enable for video pipeline Z	0b0: Software override disabled 0b1: Software override enabled
soft_bppz	4:0	Software override of BPP on video pipeline Z	0bXXXXXX: Software override value

FRONTTOP\_24 (0x320)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	soft_vc <sub>Z</sub> [1:0]	—	—	—	—	—	
<b>Reset</b>	—	—	0b00	—	—	—	—	—	
<b>Access Type</b>	—	—	Write, Read	—	—	—	—	—	
BITFIELD	BITS	DESCRIPTION				DECODE			
soft_vc <sub>Z</sub>	5:4	Virtual channel software override for video pipeline Z				0bXX: Software override value			

FRONTTOP\_27 (0x323)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	soft_dt <sub>Z</sub> [5:0]				—	—	
<b>Reset</b>	—	—	0b110000				—	—	
<b>Access Type</b>	—	—	Write, Read				—	—	
BITFIELD	BITS	DESCRIPTION				DECODE			
soft_dt <sub>Z</sub>	5:0	Datatype software override for video channel Z				0bXXXXXX: Software override value			

FRONTTOP\_29 (0x325)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	FORCE_STA RT_MIPI_FR ONTTOP	—	—	—	—	—	—	—	
<b>Reset</b>	0b0	—	—	—	—	—	—	—	
<b>Access Type</b>	Write, Read	—	—	—	—	—	—	—	
BITFIELD	BITS	DESCRIPTION				DECODE			
FORCE_STA RT_MIPI_FR ONTTOP	7	Force the MIPI receiver start without waiting for the GMSL link lock.				0b0: Do not force MIPI receiver to start without waiting for GMSL link lock 0b1: Force MIPI receiver to start without waiting for GMSL link lock			

MIPI\_RX0 (0x330)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	mipi_noncont_clk_en	ctrl1_vc_map_en	—	mipi_rx_res et	RSVD[2:0]			
<b>Reset</b>	—	0b0	0b0	—	0b0	0b000			
<b>Access Type</b>	—	Write, Read	Write, Read	—	Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE			
mipi_noncont_clk_en	6	MIPI non-continuous clock enable				0x0: enable MIPI continuous clock 0x1: enable MIPI non-continuous clock			

BITFIELD	BITS	DESCRIPTION	DECODE
ctrl1_vc_map_en	5	Virtual channel mapping enable. When enabled, the incoming virtual channel numbers are remapped according to the values in the ctrl1_vc_map0 through ctrl1_vc_map15 register bits.	0x0: Disable virtual channel mapping 0x1: Enable virtual channel mapping
mipi_rx_reset	3	Reset MIPI RX receiver (MIPI PHY). This bit is not self-clearing.	0b0: Do not reset MIPI RX 0b1: Reset MIPI RX

### MIPI\_RX1 (0x331)

BIT	7	6	5	4	3	2	1	0		
Field	ctrl1_vcxe_n	ctrl1_deske_wen	ctrl1_num_lanes[1:0]		–	–	–	–		
Reset	0b0	0b0	0b11		–	–	–	–		
Access Type	Write, Read	Write, Read	Write, Read		–	–	–	–		
BITFIELD	BITS	DESCRIPTION			DECODE					
ctrl1_vcxe_n	7	Enable the extended Virtual Channels feature			0b0: Extended VC disabled 0b1: Extended VC enabled					
ctrl1_deske_wen	6	Enable the deskew calibration for 1.5Gbps and above			0x0: Deskew calibration disabled 0x1: Deskew calibration enabled					
ctrl1_num_lanes	5:4	Select number of data lanes			0b00: One data lane 0b01: Two data lanes 0b10: Three data lanes 0b11: Four data lanes					

### MIPI\_RX2 (0x332)

BIT	7	6	5	4	3	2	1	0
Field	phy1_lane_map[3:0]					–	–	–
Reset	0xE					–	–	–
Access Type	Write, Read					–	–	–
BITFIELD	BITS	DESCRIPTION			DECODE			
phy1_lane_map	7:4	Serializer lane mapping for MIPI data lane 2 and 3. Bit[5:4] controls data lane 2. Bit[7:6] controls data lane 3. Works with phy2_lane_map register. Lane maps must be exclusive.			0bXX00: Map Sensor Lane 0 to Serializer Lane 2 0bXX01: Map Sensor Lane 1 to Serializer Lane 2 0bXX10: Map Sensor Lane 2 to Serializer Lane 2 0bXX11: Map Sensor Lane 3 to Serializer Lane 2 0b00XX: Map Sensor Lane 0 to Serializer Lane 3 0b01XX: Map Sensor Lane 1 to Serializer Lane 3 0b10XX: Map Sensor Lane 2 to Serializer Lane 3 0b11XX: Map Sensor Lane 3 to Serializer Lane 3			

MIPI\_RX3 (0x333)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	—	—	phy2_lane_map[3:0]				
<b>Reset</b>	—	—	—	—	0x4				
<b>Access Type</b>	—	—	—	—	Write, Read				
BITFIELD	BITS	DESCRIPTION					DECODE		
phy2_lane_map	3:0	Serializer lane mapping for MIPI data lane 0 and 1. Bit[1:0] controls data lane 0, Bit[3:2] controls data lane 1. Works with phy1_lane_map register. Lane maps must be exclusive.					0bXX00: Map Sensor Lane 0 to Serializer Lane 0 0bXX01: Map Sensor Lane 1 to Serializer Lane 0 0bXX10: Map Sensor Lane 2 to Serializer Lane 0 0bXX11: Map Sensor Lane 3 to Serializer Lane 0 0b00XX: Map Sensor Lane 0 to Serializer Lane 1 0b01XX: Map Sensor Lane 1 to Serializer Lane 1 0b10XX: Map Sensor Lane 2 to Serializer Lane 1 0b11XX: Map Sensor Lane 3 to Serializer Lane 1		

MIPI\_RX4 (0x334)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	phy1_pol_map[2:0]					—	—
<b>Reset</b>	—	0b000					—	—
<b>Access Type</b>	—	Write, Read					—	—
BITFIELD	BITS	DESCRIPTION					DECODE	
phy1_pol_map	6:4	Serializer lane polarity setting for MIPI data lane 2 and 3					0bXX0: Normal Polarity for data lane 2 0bXX1: Inverse Polarity for data lane 2 0bX0X: Normal Polarity for data lane 3 0bX1X: Inverse Polarity for data lane 3 0b0XX: Reserved 0b1XX: Reserved	

MIPI\_RX5 (0x335)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	phy2_pol_map[2:0]		
<b>Reset</b>	—	—	—	—	—	0b000		
<b>Access Type</b>	—	—	—	—	—	Write, Read		
BITFIELD	BITS	DESCRIPTION					DECODE	
phy2_pol_map	2:0	Serializer lane polarity setting for MIPI data lane 0, 1 and clock lane					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	

MIPI\_RX7 (0x337)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	ctrl1_mode_	ctrl1_mode_				
			UDT	RSVD[4:0]				
<b>Reset</b>	–	–	0b0	0b00000				
<b>Access Type</b>	–	–	Write, Read					
BITFIELD	BITS	DESCRIPTION				DECODE		
ctrl1_mode_	5	UDP treatment type				0x0: treat UDP as 8 bits 0x1: treat UDP as 24 bits		

MIPI\_RX8 (0x338)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	RSVD[1:0]		t_hs_settle[1:0]		RSVD[1:0]		t_clk_settle[1:0]			
<b>Reset</b>	0b01		0b01		0b01		0b01			
<b>Access Type</b>				Write, Read			Write, Read			
BITFIELD	BITS	DESCRIPTION				DECODE				
t_hs_settle	5:4	D-PHY Mode: Set typical D-PHY hs_settle timing in ns (at 2.5Gbps)				0b00: 132 (D-PHY) 0b01: 139 (D-PHY) 0b10: 153 (D-PHY) 0b11: 166 (D-PHY)				
t_clk_settle	1:0	Set typical D-PHY Tclk_settle timing in ns				0b00: 160 0b01: 220 0b10: 286 0b11: 352				

MIPI\_RX11 (0x33B)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	–	phy1_lp_err[4:0]				
<b>Reset</b>	–	–	–	0b00000				
<b>Access Type</b>	–	–	–	Read Clears All				
BITFIELD	BITS	DESCRIPTION				DECODE		
phy1_lp_err	4:0	PHY1 LP status (D-PHY only)				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		

MIPI\_RX12 (0x33C)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	phy1_hs_err[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Clears All							
BITFIELD	BITS	DESCRIPTION			DECODE			
phy1_hs_err	7:0	PHY1 high-speed status (D-PHY only)			0bXXXXXXXX1: HS sync pattern with 1-bit error detected on data lane D0 0bXXXXXX1X: HS sync pattern with 1-bit error detected on data lane D1 0bXXXXX1XX: HS sync pattern with 2 or more bit errors detected on data lane D0 0bXXXX1XXX: HS sync pattern with 2 or more bit errors 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			

MIPI\_RX13 (0x33D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	phy2_lp_err[4:0]				
<b>Reset</b>	—	—	—	0b00000				
<b>Access Type</b>	—	—	—	Read Clears All				
BITFIELD	BITS	DESCRIPTION			DECODE			
phy2_lp_err	4:0	PHY2 LP status (D-PHY only)			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			

MIPI\_RX14 (0x33E)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	phy2_hs_err[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Clears All							

BITFIELD	BITS	DESCRIPTION	DECODE
phy2_hs_err	7:0	PHY2 high-speed status (D-PHY only)	0bXXXXXXXX1: 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

MIPI\_RX19 (0x343)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	ctrl1_csi_err_l[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Read Clears All								
BITFIELD	BITS	DESCRIPTION				DECODE			
ctrl1_csi_err_l	7:0	CSI-2 Controller Status, low byte				0bXXXXXXXX1: 1-bit ECC error detected 0bXXXXXX1X: 2-bit ECC error detected 0bXYYYYYYYXX: YYYYYY = bit position of the 1-bit error 0b1XXXXXXX: CRC error detected			

MIPI\_RX20 (0x344)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	—	—	—	ctrl1_csi_err_h[2:0]			
<b>Reset</b>	—	—	—	—	—	0b000			
<b>Access Type</b>	—	—	—	—	—	Read Clears All			
BITFIELD	BITS	DESCRIPTION				DECODE			
ctrl1_csi_err_h	2:0	CSI-2 Controller Status, high bits				0bXX1: Packets terminated early 0bX1X: Frame count error detected 0b1XX: Unsupported data type received			

MIPI\_RX21 (0x345)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map0[3:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE				
ctrl1_vc_map_0	7:4	New virtual channel for VC=0. If ctrl_vc_map_en is set to 1, any time VC=0 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel				

[MIPI\\_RX22 \(0x346\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map1[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE				
ctrl1_vc_map_1	7:4	New virtual channel for VC=1. If ctrl_vc_map_en is set to 1, any time VC=1 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel				

[MIPI\\_RX23 \(0x347\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map2[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE				
ctrl1_vc_map_2	7:4	New virtual channel for VC=2. If ctrl_vc_map_en is set to 1, any time VC=2 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel				

[MIPI\\_RX60 \(0x36C\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map3[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE				
ctrl1_vc_map_3	7:4	New virtual channel for VC=3. If ctrl_vc_map_en is set to 1, any time VC=3 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel				

MIPI\_RX61 (0x36D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map4[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	BITS	DESCRIPTION					DECODE	
ctrl1_vc_map 4	7:4	New virtual channel for VC=4. If ctrl_vc_map_en is set to 1, any time VC=4 is seen, the virtual channel number is replaced with the value in this register.					0xX: New virtual channel	

MIPI\_RX62 (0x36E)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map5[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	BITS	DESCRIPTION					DECODE	
ctrl1_vc_map 5	7:4	New virtual channel for VC=5. If ctrl_vc_map_en is set to 1, any time VC=5 is seen, the virtual channel number is replaced with the value in this register.					0xX: New virtual channel	

MIPI\_RX63 (0x36F)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map6[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	BITS	DESCRIPTION					DECODE	
ctrl1_vc_map 6	7:4	New virtual channel for VC=6. If ctrl_vc_map_en is set to 1, any time VC=6 is seen, the virtual channel number is replaced with the value in this register.					0xX: New virtual channel	

EXT00 (0x377)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map7[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	BITS	DESCRIPTION					DECODE	

BITFIELD	BITS	DESCRIPTION	DECODE
ctrl1_vc_map 7	7:4	New virtual channel for VC=7. If ctrl_vc_map_en is set to 1, any time VC=7 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel

EXT0 (0x378)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map8[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
ctrl1_vc_map 8	7:4	New virtual channel for VC=8. If ctrl_vc_map_en is set to 1, any time VC=8 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel

EXT1 (0x379)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map9[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
ctrl1_vc_map 9	7:4	New virtual channel for VC=9. If ctrl_vc_map_en is set to 1, any time VC=9 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel

EXT2 (0x37A)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map10[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	BITS	DESCRIPTION	DECODE					
ctrl1_vc_map 10	7:4	New virtual channel for VC=10. If ctrl_vc_map_en is set to 1, any time VC=10 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel					

[EXT3 \(0x37B\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map11[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	DESCRIPTION					DECODE		
ctrl1_vc_map 11	7:4	New virtual channel for VC=11. If ctrl_vc_map_en is set to 1, any time VC=11 is seen, the virtual channel number is replaced with the value in this register.					0XX: New virtual channel	

[EXT4 \(0x37C\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map12[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	DESCRIPTION					DECODE		
ctrl1_vc_map 12	7:4	New virtual channel for VC=12. If ctrl_vc_map_en is set to 1, any time VC=12 is seen, the virtual channel number is replaced with the value in this register.					0XX: New virtual channel	

[EXT5 \(0x37D\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map13[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	DESCRIPTION					DECODE		
ctrl1_vc_map 13	7:4	New virtual channel for VC=13. If ctrl_vc_map_en is set to 1, any time VC=13 is seen, the virtual channel number is replaced with the value in this register.					0XX: New virtual channel	

[EXT6 \(0x37E\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map14[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–
BITFIELD	DESCRIPTION					DECODE		

BITFIELD	BITS	DESCRIPTION	DECODE
ctrl1_vc_map_14	7:4	New virtual channel for VC=14. If ctrl_vc_map_en is set to 1, any time VC=14 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel

EXT7 (0x37F)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_vc_map15[3:0]					–	–	–
<b>Reset</b>	0x0					–	–	–
<b>Access Type</b>	Write, Read					–	–	–

BITFIELD	BITS	DESCRIPTION	DECODE
ctrl1_vc_map_15	7:4	New virtual channel for VC=15. If ctrl_vc_map_en is set to 1, any time VC=15 is seen, the virtual channel number is replaced with the value in this register.	0xX: New virtual channel

EXT8 (0x380)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	RSVD[1:0]			RSVD[1:0]		RSVD	RSVD
<b>Reset</b>	0b0	0b0			0b0		0b0	0b0
<b>Access Type</b>							Read Only	

BITFIELD	BITS	DESCRIPTION	DECODE
tun_fifo_overflow	0	Tunnel FIFO overflow. Tunnel mode only.	0x0: No overflow 0x1: Overflow error triggered

EXT9 (0x381)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>								

EXT11 (0x383)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	Tun_Mode	RSVD	–	–	RSVD	RSVD	RSVD[1:0]		
<b>Reset</b>	0x1	0x0	–	–	0x0	0x0	0x0		
<b>Access Type</b>	Write, Read								

BITFIELD	BITS	DESCRIPTION	DECODE
Tun_Mode	7	Select Tunnel mode	0x0: Select Pixel mode 0x1: Select Tunnel mode

[EXT21 \(0x38D\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	phy1_pkt_cnt[7:0]							
<b>Reset</b>	0b0							
<b>Access Type</b>	Read Only							
BITFIELD	BITS	DESCRIPTION						DECODE
phy1_pkt_cnt	7:0	MIPI PHY1 Packets Received						0xXX: Packets Received

[EXT22 \(0x38E\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	csi1_pkt_cnt[7:0]							
<b>Reset</b>	0b0							
<b>Access Type</b>	Read Only							
BITFIELD	BITS	DESCRIPTION						DECODE
csi1_pkt_cnt	7:0	MIPI Controller 1 Packets Processed						0xXX: Packets Processed

[EXT23 \(0x38F\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	tun_pkt_cnt[7:0]							
<b>Reset</b>	0b0							
<b>Access Type</b>	Read Only							
BITFIELD	BITS	DESCRIPTION						DECODE
tun_pkt_cnt	7:0	MIPI Tunnel Packets Processed						0xXX: Packets Processed

[EXT24 \(0x390\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	phy_clk_cnt[7:0]							
<b>Reset</b>	0b0							
<b>Access Type</b>	Read Only							
BITFIELD	BITS	DESCRIPTION						DECODE
phy_clk_cnt	7:0	MIPI RX Clock Received. The changing value indicates MIPI clock lane is running.						0xXX: MIPI RX Clock Count

FRONTTOP\_EXT8 (0x3C8)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	mem_dt3_selz[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION				DECODE		
mem_dt3_selz	7:0	Select a designated datatype to route to video pipeline Z for VS. Bits 7:6 are for selecting the two LSBs of the virtual channel. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt3_selz_en and mem_dt4_selz_en are 0.				0b00XXXXXX: VC0,4,8,12, Designated datatype 0b01XXXXXX: VC1,5,9,13, Designated datatype 0b10XXXXXX: VC2,6,10,14, Designated datatype 0b11XXXXXX: VC3,7,11,15, Designated datatype		

FRONTTOP\_EXT9 (0x3C9)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	mem_dt4_selz[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION				DECODE		
mem_dt4_selz	7:0	Select a designated datatype to route to video pipeline Z for VS. Bits 7:6 are for selecting the two LSBs of the virtual channel. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt3_selz_en and mem_dt4_selz_en are 0.				0b00XXXXXX: VC0,4,8,12, Designated datatype 0b01XXXXXX: VC1,5,9,13, Designated datatype 0b10XXXXXX: VC2,6,10,14, Designated datatype 0b11XXXXXX: VC3,7,11,15, Designated datatype		

FRONTTOP\_EXT10 (0x3CA)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	mem_dt5_selz[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION				DECODE		

BITFIELD	BITS	DESCRIPTION	DECODE
mem_dt5_selz	7:0	Select a designated datatype to route to video pipeline Z for HS/DE. Bits 7:6 are for selecting the two LSBs of the virtual channel. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt5_selz_en and mem_dt6_selz_en are 0.	0b00XXXXXX: VC0,4,8,12, Designated datatype 0b01XXXXXX: VC1,5,9,13, Designated datatype 0b10XXXXXX: VC2,6,10,14, Designated datatype 0b11XXXXXX: VC3,7,11,15, Designated datatype

### FRONTTOP\_EXT11 (0x3CB)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	mem_dt6_selz[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
mem_dt6_selz	7:0	Select a designated datatype to route to video pipeline Z for HS/DE. Bits 7:6 are for selecting the two LSBs of the virtual channel. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt5_selz_en and mem_dt6_selz_en are 0.						0b00XXXXXX: VC0,4,8,12, Designated datatype 0b01XXXXXX: VC1,5,9,13, Designated datatype 0b10XXXXXX: VC2,6,10,14, Designated datatype 0b11XXXXXX: VC3,7,11,15, Designated datatype

### FRONTTOP\_EXT17 (0x3D1)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	mem_dt6_s elz_en	mem_dt5_s elz_en	mem_dt4_s elz_en	mem_dt3_s elz_en
<b>Reset</b>	—	—	—	—	0b0	0b0	0b0	0b0
<b>Access Type</b>	—	—	—	—	Write, Read	Write, Read	Write, Read	Write, Read
BITFIELD	BITS	DESCRIPTION						DECODE
mem_dt6_selz_en	3	Enable datatype designated in mem_dt6_selz to route to video pipeline Z. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt5_selz_en and mem_dt6_selz_en are 0.						0b0: Disable 0b1: Enable
mem_dt5_selz_en	2	Enable datatype designated in mem_dt5_selz to route to video pipeline Z. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt5_selz_en and mem_dt6_selz_en are 0.						0b0: Disable 0b1: Enable

BITFIELD	BITS	DESCRIPTION	DECODE
mem_dt4_selz_en	1	Enable datatype designated in mem_dt4_selz to route to video pipeline Z. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt3_selz_en and mem_dt4_selz_en are 0.	0b0: Disable 0b1: Enable
mem_dt3_selz_en	0	Enable datatype designated in mem_dt3_selz to route to video pipeline Z. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt3_selz_en and mem_dt4_selz_en are 0.	0b0: Disable 0b1: Enable

**EXTA (0x3DC)**

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	mem_dt7_selz[6:0]						
<b>Reset</b>	—	0b0000000						
<b>Access Type</b>	—	Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
mem_dt7_selz	6:0	Select designated datatype to route to video pipeline Z. Bit 6 is the enable. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt1_selz[6], mem_dt2_selz[6], mem_dt7_selz[6], and mem_dt8_selz[6] are all 0.	0b0XXXXXX: Datatype selection disabled 0b1XXXXXX: Datatype enabled for datatype selected to route to video pipeline

**EXTB (0x3DD)**

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	mem_dt8_selz[6:0]						
<b>Reset</b>	—	0b0000000						
<b>Access Type</b>	—	Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
mem_dt8_selz	6:0	Select designated datatype to route to video pipeline Z. Bit 6 is the enable. Bits 5:0 are for the datatype. Used for filtering which datatypes are routed to video pipeline Z. If no filtering is enabled, then all datatypes are routed. This happens when mem_dt1_selz[6], mem_dt2_selz[6], mem_dt7_selz[6], and mem_dt8_selz[6] are all 0.	0b0XXXXXX: Datatype selection disabled 0b1XXXXXX: Datatype enabled for datatype selected to route to video pipeline

VTX0 (0x3E0)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	—	VS_TRIGGER	REF_VTG_MODE[1:0]		HS_INV	GEN_HS	VS_INV	GEN_VS		
<b>Reset</b>	—	0b1	0b11		0b0	0b0	0b0	0b0		
<b>Access Type</b>	—	Write, Read	Write, Read		Write, Read	Write, Read	Write, Read	Write, Read		
BITFIELD	BITS	DESCRIPTION				DECODE				
VS_TRIGGER	6	Select VS trigger edge (positive vs. negative polarity of VS)				0b0: Falling edge is VS trigger 0b1: Rising edge is VS trigger				
REF_VTG_MODE	5:4	Selects one of the following modes for video interface timing generation. Used when REF_VTG GEN_HS or GEN_VS are enabled.								
		VS tracking mode. VS input's period (VS_HIGH + VS_LOW) is tracked. After VS tracking is locked, any VS input edge (glitches) not in the expected PCLK cycle is ignored. VS tracking is locked with three consecutive matches and unlocked by three consecutive mismatches. When unlocked or power-up, the next VS input edge is assumed to be the right VS edge.								
		VS one-trigger mode. One VS input edge triggers the generation of one frame of VSO/HSO/DEO. If the next VS input edge comes earlier or later than expected by VS period, the newly generated frame is correct. The current VSO/HSO/DEO is cut or extended at the point of the rising edge of the newly generated VSO/HSO/DEO.				0b00: VS tracking mode 0b01: VS one trigger mode 0b10: Auto repeat mode 0b11: VS tracking mode				
HS_INV	3	Invert HS output of video timing generator				0b0: Do not invert HS 0b1: Invert HS				
GEN_HS	2	Enable generation of HS output				0b0: Disable generation of HS 0b1: Enable generation of HS				
VS_INV	1	Invert VS output of video timing generator				0b0: Do not invert VS 0b1: Invert VS				
GEN_VS	0	Enable generation of VS output				0b0: Disable generation of VS 0b1: Enable generation of VS				

VTX1 (0x3E1)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_HIGH_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
VS_HIGH_2	7:0	VS High Period in terms of PCLK cycles (Bits [23:16])			0xXX: VS high period high byte			

VTX2 (0x3E2)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_HIGH_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
VS_HIGH_1	7:0	VS High Period in terms of PCLK cycles (Bits [15:8])			0xXX: VS high period middle byte			

VTX3 (0x3E3)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_HIGH_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
VS_HIGH_0	7:0	VS High Period in terms of PCLK cycles (Bits [7:0])			0xXX: VS high period low byte			

VTX4 (0x3E4)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_LOW_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
VS_LOW_2	7:0	VS Low Period in terms of PCLK cycles (Bits [23:16])			0xXX: VS low period high byte			

VTX5 (0x3E5)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	VS_LOW_1[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
VS_LOW_1	7:0	VS Low Period in terms of PCLK cycles (Bits [15:8])				0xXX: VS low period middle byte			

VTX6 (0x3E6)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	VS_LOW_0[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
VS_LOW_0	7:0	VS Low Period in terms of PCLK cycles (Bits [7:0])				0xXX: VS low period low byte			

VTX7 (0x3E7)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	V2H_2[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
V2H_2	7:0	Horizontal sync delay. VS edge to the rising edge of the first HS in terms of PCLK cycles (Bits [23:16])				0xXX: HS delay high byte			

VTX8 (0x3E8)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	V2H_1[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	DESCRIPTION								DECODE
V2H_1	7:0	Horizontal sync delay. VS edge to the rising edge of the first HS in terms of PCLK cycles (Bits [15:8])				0xXX: HS delay middle byte			

VTX9 (0x3E9)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	V2H_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
V2H_0	7:0	Horizontal sync delay. VS edge to the rising edge of the first HS in terms of PCLK cycles (Bits [7:0])			0xXX: HS delay low byte			

VTX10 (0x3EA)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_HIGH_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_HIGH_1	7:0	HS High Period in terms of PCLK cycles (Bits [15:8])			0xXX: HS high period high byte			

VTX11 (0x3EB)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_HIGH_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_HIGH_0	7:0	HS High Period in terms of PCLK cycles (Bits [7:0])			0xXX: HS high period low byte			

VTX12 (0x3EC)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_LOW_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_LOW_1	7:0	HS Low Period in terms of PCLK cycles (Bits [15:8])			0xXX: HS low period high byte			

VTX13 (0x3ED)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_LOW_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_LOW_0	7:0	HS Low Period in terms of PCLK cycles (Bits [7:0])			0xXX: HS low period low byte			

VTX14 (0x3EE)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_CNT_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_CNT_1	7:0	Number of HS pulses per frame (Bits [15:8])			0xXX: HS pulses per frame high byte			

VTX15 (0x3EF)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	HS_CNT_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
HS_CNT_0	7:0	Number of HS pulses per frame (Bits [7:0])			0xXX: HS pulses per frame low byte			

REF\_VTG0 (0x3F0)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFGEN_L_OCKED	REFGEN_PREDEF_EN	REFGEN_PREDEF_FRE	REFGEN_PREDEF_FRE_EQ_ALT	–	REFGEN_RST	REFGEN_EN	–
<b>Reset</b>	0b0	0b1	0b01	0b0	–	0b0	0b0	0b0
<b>Access Type</b>	Read Only	Write, Read	Write, Read	Write, Read	–	Write, Read	Write, Read	–
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
REFGEN_LOCKED	7	Reference generation PLL is locked (for DPLL_OUT)			0b0: Reference generation PLL not locked 0b1: Reference generation PLL locked			
REFGEN_PREDEF_EN	6	Enable predefined clock settings for reference generation PLL			0b0: Disable pre-defined clock settings for reference generation PLL 0b1: Enable pre-defined clock settings for reference generation PLL			

BITFIELD	BITS	DESCRIPTION	DECODE
REFGEN_P REDEF_FRE Q	5:4	Predefined reference generation PLL frequency setting Set REFGEN_PREDEF_FREQ_ALT=1 to select alternative frequency selections	0x0: 19.2 MHz / 13.5MHz (ALT) 0x1: 27.0 MHz / 24MHz (ALT) 0x2: 37.125 MHz / Reserved (ALT) 0x3: 74.25 MHz / Reserved (ALT)
REFGEN_P REDEF_FRE Q_ALT	3	Enable alternative predefined reference generation PLL frequency setting	0x0: Original table 0x1: Alternative table
REFGEN_R ST	1	Reset reference generation PLL	0b0: Do not reset reference generation PLL 0b1: Reset reference generation PLL
REFGEN_E N	0	Enable reference generation PLL	0b0: Disable reference generation PLL 0b1: Enable reference generation PLL

[REF\\_VTG4 \(0x3F4\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFGEN_FB_FRACT_L[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
REFGEN_FB _FRACT_L	7:0	Reference generator PLL feedback divider fraction value when predefined mode is disabled (REFGEN_PREDEF_EN=0)			0xXX: Low byte of reference generator PLL feedback divider fraction value			

[REF\\_VTG5 \(0x3F5\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFGEN_FB_FRACT_H[3:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
REFGEN_FB _FRACT_H	3:0	Reference generator PLL feedback divider fraction value when predefined mode is disabled (REFGEN_PREDEF_EN=0)			0X: High nibble of reference generator PLL feedback divider fraction value			

[REF\\_VTG6 \(0x3F6\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_DLY_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
VS_DLY_2	7:0	VS Delay in terms of pixel clock cycles. The output VS is delayed by VS_DELAY cycles from the input VS. (Bits [23:16])			0XX: VS delay high byte			

[REF\\_VTG7 \(0x3F7\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_DLY_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
VS_DLY_1	7:0	VS Delay in terms of pixel clock cycles. The output VS is delayed by VS_DELAY cycles from the input VS. (Bits [15:8])			0xXX: VS delay middle byte			

[REF\\_VTG8 \(0x3F8\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VS_DLY_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
VS_DLY_0	7:0	VS Delay in terms of pixel clock cycles. The output VS is delayed by VS_DELAY cycles from the input VS. (Bits [7:0])			0xXX: VS delay low byte			

[REF\\_VTG9 \(0x3F9\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REF_VTG_TRIG_EN	—	—	REF_VTG_TRIG_ID[4:0]				
<b>Reset</b>	0b0	—	—	0b11110				
<b>Access Type</b>	Write, Read	—	—	Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE			
REF_VTG_TRIG_EN	7	Enable receiving REF VTG trigger signal			0b0: Disable reception of REF VTG input signal 0b1: Enable reception of REF VTG input signal			
REF_VTG_TRIG_ID	4:0	GPIO ID used for receiving REF_VTG_TRIGGER			0bXXXXX: GPIO ID selected for receiving REF_VTG_TRIGGER			

[ADC\\_CTRL\\_0 \(0x500\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	buf_bypass	RSVD	RSVD	adc_chgpu_mp_pu	adc_refbuf_pu	buf_pu	adc_pu	cpu_adc_start
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read			Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE			
buf_bypass	7	Bypass input buffer	0b0: Use input buffer 0b1: Bypass input buffer stage			
adc_chgpump_p_pu	4	ADC charge pump power up	0b0: ADC charge pump powered off 0b1: ADC charge pump powered on			
adc_refbufp_u	3	ADC reference buffer power up	0b0: ADC reference buffer powered off 0b1: ADC reference buffer powered on			
buf_pu	2	ADC input buffer power up	0b0: ADC input buffer powered off 0b1: ADC input buffer powered on			
adc_pu	1	ADC power up	0b0: ADC powered off 0b1: ADC powered on			
cpu_adc_start	0	Start ADC conversion. Bit is automatically cleared to zero after completion of the conversion.	0b0: Conversion complete 0b1: Set to 1 to start ADC conversion			

### ADC\_CTRL\_1 (0x501)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	adc_chsel[3:0]				adc_clk_en	adc_refsel	adc_scale	RSVD	
<b>Reset</b>	0x0				0b0	0b0	0b0	0b0	
<b>Access Type</b>	Write, Read				Write, Read	Write, Read	Write, Read		
BITFIELD	BITS	DESCRIPTION				DECODE			
adc_chsel	7:4	ADC channel select. Selects ADC input to be converted (see inmux_en bit).				0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: Reserved 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved			
adc_clk_en	3	ADC clock enable. Must be enabled to activate ADC.				0b0: ADC clock disable 0b1: ADC clock enable			
adc_refsel	2	ADC reference voltage select				0b0: Select internal bandgap voltage (or external V <sub>REF</sub> pin) as ADC reference voltage (see adc_xref bit) 0b1: Select V <sub>DD18</sub> /2 as ADC reference voltage			
adc_scale	1	ADC scale				0b0: Normal operation 0b1: Scale ADC input down by 1/2			

[ADC\\_CTRL\\_2 \(0x502\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	—	RSVD	RSVD	adc_div[1:0]	adc_xref	lnmux_en	
<b>Reset</b>	0b0	—	0b0	0b0	0b00	0b0	0b0	
<b>Access Type</b>		—			Write, Read	Write, Read	Write, Read	
BITFIELD	BITS	DESCRIPTION					DECODE	
adc_div	3:2	ADC[2:0] internal divider setting					0b00: Divide ADC input voltage by 1 0b01: Divide ADC input voltage by 2 0b10: Divide ADC input voltage by 3 0b11: Divide ADC input voltage by 4	
adc_xref	1	Enable use of ADC external reference					0b0: Use internal 1.25V voltage reference for ADC 0b1: Use external voltage reference connected to V <sub>REF</sub> pin for ADC	
lnmux_en	0	Enable the input mux to the ADC to allow for a break before make connection sequence					0b0: Input mux is disconnected 0b1: Input mux is connected per adc_chsel field	

[ADC\\_DATA0 \(0x508\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	adc_data_l[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS	DESCRIPTION					DECODE	
adc_data_l	7:0	Lower byte of 10-bit ADC converted sample data output					0xX: Lower byte of 10-bit ADC converted value	

[ADC\\_DATA1 \(0x509\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	bypass_volt temp_corr	—	—	—	—	—	adc_data_h[1:0]	
<b>Reset</b>	0b0	—	—	—	—	—	0b00	
<b>Access Type</b>	Write, Read	—	—	—	—	—	Read Only	
BITFIELD	BITS	DESCRIPTION					DECODE	
bypass_volt temp_corr	7	Bypass the ADC calibration adjustment. Use this to get raw ADC output. Set this to 1 if using ADC external reference and the adc_xref bit is set to 1.					0x0: Do not bypass 0x1: Bypass	
adc_data_h	1:0	Upper 2-bits of 10-bit ADC converted sample data output					0xX: Upper 2-bits of 10-bit ADC converted value	

ADC\_INTRIE0 (0x50C)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	adc_calDone_ie	adc_overRange_ie	adc_tmon_cal_ood_ie	RSVD	adc_lo_limit_ie	adc_hi_limit_ie	adc_ref_ready_ie	adc_done_ie
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read	Write, Read	Write, Read		Write, Read	Write, Read	Write, Read	Write, Read
<b>BITFIELD</b>	<b>BITS</b>	<b>DESCRIPTION</b>						<b>DECODE</b>
adc_calDone_ie	7	Signal that ADC accuracy/temperature sensor calibration is complete, enable. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
adc_overRange_ie	6	ADC Digital Correction Overrange enabled. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
adc_tmon_cali_ood_ie	5	Enable temperature sensor out-of-date interrupt. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
adc_lo_limit_ie	3	Enable ADC low limit monitor interrupt. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
adc_hi_limit_ie	2	Enable ADC high limit monitor interrupt. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
adc_ref_ready_ie	1	Enable ADC ready interrupt. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
adc_done_ie	0	Enable ADC Conversion Done Interrupt. Need to also set ADC_INT_OEN.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled

ADC\_INTRIE1 (0x50D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ch7_hi_limit_ie	ch6_hi_limit_ie	ch5_hi_limit_ie	ch4_hi_limit_ie	ch3_hi_limit_ie	ch2_hi_limit_ie	ch1_hi_limit_ie	ch0_hi_limit_ie
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read
<b>BITFIELD</b>	<b>BITS</b>	<b>DESCRIPTION</b>						<b>DECODE</b>
ch7_hi_limit_ie	7	Enable Channel 7 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch6_hi_limit_ie	6	Enable Channel 6 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch5_hi_limit_ie	5	Enable Channel 5 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch4_hi_limit_ie	4	Enable Channel 4 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch3_hi_limit_ie	3	Enable Channel 3 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.						0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled

BITFIELD	BITS	DESCRIPTION	DECODE
ch2_hi_limit_ie	2	Enable Channel 2 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch1_hi_limit_ie	1	Enable Channel 1 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch0_hi_limit_ie	0	Enable Channel 0 high limit monitor interrupt. Need to also set ADC_INT_OEN and adc_hi_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled

### ADC\_INTRIE2 (0x50E)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ch7_lo_limit_ie	ch6_lo_limit_ie	ch5_lo_limit_ie	ch4_lo_limit_ie	ch3_lo_limit_ie	ch2_lo_limit_ie	ch1_lo_limit_ie	ch0_lo_limit_ie
<b>Reset</b>	0b0							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
ch7_lo_limit_ie	7	Enable Channel 7 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch6_lo_limit_ie	6	Enable Channel 6 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch5_lo_limit_ie	5	Enable Channel 5 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch4_lo_limit_ie	4	Enable Channel 4 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch3_lo_limit_ie	3	Enable Channel 3 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch2_lo_limit_ie	2	Enable Channel 2 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch1_lo_limit_ie	1	Enable Channel 1 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled
ch0_lo_limit_ie	0	Enable Channel 0 low limit monitor interrupt. Need to also set ADC_INT_OEN and adc_lo_limit_ie.	0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled

[ADC\\_INTRIE3 \(0x50F\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	reflim_ie	reflimscl1_ie	reflimscl2_ie	reflimscl3_ie	RSVD	tmon_err_ie	RSVD
<b>Reset</b>	—	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	—	Write, Read	Write, Read	Write, Read	Write, Read		Write, Read	
BITFIELD	BITS	DESCRIPTION				DECODE		
reflim_ie	6	Enable the REFLIM interrupt (for ADC BIST)				0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled		
reflimscl1_ie	5	Enable the REFLIMSCL1 interrupt (for ADC BIST)				0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled		
reflimscl2_ie	4	Enable the REFLIMSCL2 interrupt (for ADC BIST)				0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled		
reflimscl3_ie	3	Enable the REFLIMSCL3 interrupt (for ADC BIST)				0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled		
tmon_err_ie	1	Enable the temperature sensor error interrupt				0b0: ADC interrupt source disabled 0b1: ADC interrupt source enabled		

[ADC\\_INTR0 \(0x510\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	adc_calDone_if	adc_overRange_if	adc_tmon_calibration_ood_if	RSVD	adc_lo_limit_if	adc_hi_limit_if	adc_ref_ready_if	adc_done_if
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Read Clears All	Read Clears All	Read Clears All		Read Clears All	Read Clears All	Read Clears All	Read Clears All
BITFIELD	BITS	DESCRIPTION				DECODE		
adc_calDone_if	7	ADC accuracy/temperature sensor done flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set		
adc_overRange_if	6	Detected that ADC input voltage exceeds ADC valid input range flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set		
adc_tmon_calibration_ood_if	5	Temperature sensor calibration has expired and should be rerun flag (see run_tmoncal bit). Cleared when read.				0b0: Flag cleared 0b1: Flag set		
adc_lo_limit_if	3	ADC low limit monitor interrupt flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set		
adc_hi_limit_if	2	ADC high limit monitor interrupt flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set		
adc_ref_ready_if	1	After powerup the ADC is ready to be used flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set		
adc_done_if	0	ADC conversion done interrupt flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set		

ADC\_INTR1 (0x511)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ch7_hi_limit_if	ch6_hi_limit_if	ch5_hi_limit_if	ch4_hi_limit_if	ch3_hi_limit_if	ch2_hi_limit_if	ch1_hi_limit_if	ch0_hi_limit_if
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All
BITFIELD	BITS	<b>DESCRIPTION</b>						<b>DECODE</b>
ch7_hi_limit_if	7	Channel 7 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch6_hi_limit_if	6	Channel 6 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch5_hi_limit_if	5	Channel 5 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch4_hi_limit_if	4	Channel 4 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch3_hi_limit_if	3	Channel 3 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch2_hi_limit_if	2	Channel 2 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch1_hi_limit_if	1	Channel 1 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch0_hi_limit_if	0	Channel 0 high limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set

ADC\_INTR2 (0x512)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ch7_lo_limit_if	ch6_lo_limit_if	ch5_lo_limit_if	ch4_lo_limit_if	ch3_lo_limit_if	ch2_lo_limit_if	ch1_lo_limit_if	ch0_lo_limit_if
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All	Read Clears All
BITFIELD	BITS	<b>DESCRIPTION</b>						<b>DECODE</b>
ch7_lo_limit_if	7	Channel 7 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch6_lo_limit_if	6	Channel 6 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch5_lo_limit_if	5	Channel 5 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch4_lo_limit_if	4	Channel 4 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch3_lo_limit_if	3	Channel 3 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch2_lo_limit_if	2	Channel 2 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set
ch1_lo_limit_if	1	Channel 1 low limit monitor interrupt flag. Cleared when read.						0b0: Flag cleared 0b1: Flag set

BITFIELD	BITS	DESCRIPTION				DECODE			
ch0_lo_limit_if	0	Channel 0 low limit monitor interrupt flag. Cleared when read.				0b0: Flag cleared 0b1: Flag set			

### [ADC\\_INTR3 \(0x513\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	reflim_if	reflimscl1_if	reflimscl2_if	reflimscl3_if	RSVD	tmon_err_if	RSVD
<b>Reset</b>	—	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	—	Read Clears All	Read Clears All	Read Clears All	Read Clears All		Read Clears All	

BITFIELD	BITS	DESCRIPTION				DECODE			
reflim_if	6	Returned value from ADC BIST test (divided by 1 input mode) exceeds test limit set by REFLIM register. Cleared when read.				0b0: Flag cleared 0b1: Flag set			
reflimscl1_if	5	Returned value from ADC BIST test (divided by 2 input mode) exceeds test limit set by REFLIMSCL1 register. Cleared when read.				0b0: Flag cleared 0b1: Flag set			
reflimscl2_if	4	Returned value from ADC BIST test (divided by 4 input mode) exceeds test limit set by REFLIMSCL2 register. Cleared when read.				0b0: Flag cleared 0b1: Flag set			
reflimscl3_if	3	Returned value from ADC BIST test (divided by 8 input mode) exceeds test limit set by REFLIMSCL3 register. Cleared when read.				0b0: Flag cleared 0b1: Flag set			
tmon_err_if	1	Discrepancy between temperature sensor and redundant temperature sensor exceeds test limit set by TLIMIT register.				0b0: Flag cleared 0b1: Flag set			

### [ADC\\_LIMIT0\\_0 \(0x514\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	chLoLimit_I0[7:0]								
<b>Reset</b>	0x00								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION				DECODE			
chLoLimit_I0	7:0	ADC Output Register set 0 - low limit threshold value, low bits				0XX: Low limit low byte			

### [ADC\\_LIMIT0\\_1 \(0x515\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	chHiLimit_I0[3:0]								
<b>Reset</b>	0xF								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION				DECODE			
chHiLimit_I0	7:4	ADC Output Register set 0 - high limit threshold value, low bits				0bX: High limit low nibble			

BITFIELD	BITS	DESCRIPTION	DECODE
chLoLimit_h0	1:0	ADC Output Register set 0 - low limit threshold value, high bits	0bXX: Low limit high bits

[ADC\\_LIMIT0\\_2 \(0x516\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h0[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
chHiLimit_h0	5:0	ADC Output Register set 0 - high limit threshold value, high bits	0bXXXXXX: High limit high bits

[ADC\\_LIMIT0\\_3 \(0x517\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	div_sel0[1:0]					
<b>Reset</b>	—	—	0b00					
<b>Access Type</b>	—	—	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
div_sel0	5:4	ADC channel 0 divider setting	0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4
ch_sel0	3:0	ADC Input Select for ADC Output Register set 0	0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved

[ADC\\_LIMIT1\\_0 \(0x518\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chLoLimit_l1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
chLoLimit_l1	7:0	ADC Output Register set 1 - low limit threshold value, low bits	0xXX: Low limit low byte

#### ADC\_LIMIT1\_1 (0x519)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>		chHiLimit_l1[3:0]			–	–	chLoLimit_h1[1:0]	
<b>Reset</b>		0xF			–	–	0b00	
<b>Access Type</b>		Write, Read			–	–	Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
chHiLimit_l1	7:4	ADC Output Register set 1 - high limit threshold value, low bits	0bX: High limit low nibble
chLoLimit_h1	1:0	ADC Output Register set 1 - low limit threshold value, high bits	0bXX: Low limit high bits

#### ADC\_LIMIT1\_2 (0x51A)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	–	–	–	–	–	–	chHiLimit_h1[5:0]		
<b>Reset</b>	–	–	–	–	–	–	0b111111		
<b>Access Type</b>	–	–	–	–	–	–	Write, Read		
BITFIELD	BITS	DESCRIPTION	DECODE						
chHiLimit_h1	5:0	ADC Output Register set 1 - high limit threshold value, high bits	0bXXXXXX: High limit high bits						

#### ADC\_LIMIT1\_3 (0x51B)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	–	–	div_sel1[1:0]	–	–	–	ch_sel1[3:0]		
<b>Reset</b>	–	–	0b00	–	–	–	0x3		
<b>Access Type</b>	–	–	–	Write, Read	–	–	Write, Read		
BITFIELD	BITS	DESCRIPTION	DECODE						
div_sel1	5:4	ADC channel 1 divider setting	0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4						

BITFIELD	BITS	DESCRIPTION	DECODE
ch_sel1	3:0	ADC Input Select for ADC Output Register set 1	0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved

[ADC\\_LIMIT2\\_0 \(0x51C\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chLoLimit_I2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
chLoLimit_I2	7:0	ADC Output Register set 2 - low limit threshold value, low bits						0xXX: Low limit low byte

[ADC\\_LIMIT2\\_1 \(0x51D\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chHiLimit_I2[3:0]							
<b>Reset</b>	0xF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
chHiLimit_I2	7:4	ADC Output Register set 2 - high limit threshold value, low bits						0bX: High limit low nibble
chLoLimit_h2	1:0	ADC Output Register set 2 - low limit threshold value, high bits						0bXX: Low limit high bits

[ADC\\_LIMIT2\\_2 \(0x51E\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h2[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
chHiLimit_h2	5:0	ADC Output Register set 2 - high limit threshold value, high bits	0bXXXXXX: High limit high bits

### ADC\_LIMIT2\_3 (0x51F)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	div_sel2[1:0]		ch_sel2[3:0]			
<b>Reset</b>	—	—	0b00		0x3			
<b>Access Type</b>	—	—	Write, Read		Write, Read			

BITFIELD	BITS	DESCRIPTION	DECODE
div_sel2	5:4	ADC channel 2 divider setting	0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4
ch_sel2	3:0	ADC Input Select for ADC Output Register set 2	0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved

### ADC\_LIMIT3\_0 (0x520)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>					chLoLimit_I3[7:0]				
<b>Reset</b>					0x00				
<b>Access Type</b>					Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE			
chLoLimit_I3	7:0	ADC Output Register set 3 - low limit threshold value, low bits				0XX: Low limit low byte			

### ADC\_LIMIT3\_1 (0x521)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>			chHiLimit_I3[3:0]		—	—	chLoLimit_h3[1:0]	
<b>Reset</b>			0xF		—	—	0b00	
<b>Access Type</b>			Write, Read		—	—	Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
chHiLimit_l3	7:4	ADC Output Register set 3 - high limit threshold value, low bits	0bX: High limit low nibble
chLoLimit_h3	1:0	ADC Output Register set 3 - low limit threshold value, high bits	0bXX: Low limit high bits

### ADC\_LIMIT3\_2 (0x522)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h3[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					
BITFIELD	BITS	DESCRIPTION					DECODE	
chHiLimit_h3	5:0	ADC Output Register set 3 - high limit threshold value, high bits					0bXXXXXX: High limit high bits	

### ADC\_LIMIT3\_3 (0x523)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	div_sel3[1:0]					ch_sel3[3:0]
<b>Reset</b>	—	—	0b00					0x3
<b>Access Type</b>	—	—	Write, Read					Write, Read
BITFIELD	BITS	DESCRIPTION					DECODE	
div_sel3	5:4	ADC channel 3 divider setting					0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4	
ch_sel3	3:0	ADC Input Select for ADC Output Register set 3					0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved	

[ADC\\_LIMIT4\\_0 \(0x524\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chLoLimit_l4[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
chLoLimit_l4	7:0	ADC Output Register set 4 - low limit threshold value, low bits			0xXX: Low limit low byte			

[ADC\\_LIMIT4\\_1 \(0x525\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chHiLimit_l4[3:0]							
<b>Reset</b>	0xF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
chHiLimit_l4	7:4	ADC Output Register set 4 - high limit threshold value, low bits			0bX: High limit low nibble			
chLoLimit_h4	1:0	ADC Output Register set 4 - low limit threshold value, high bits			0bXX: Low limit high bits			

[ADC\\_LIMIT4\\_2 \(0x526\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h4[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
chHiLimit_h4	5:0	ADC Output Register set 4 - high limit threshold value, high bits			0bXXXXXX: High limit high bits			

[ADC\\_LIMIT4\\_3 \(0x527\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	div_sel4[1:0]			ch_sel4[3:0]			
<b>Reset</b>	—	—	0b00			0x3			
<b>Access Type</b>	—	—	Write, Read			Write, Read			
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>				
div_sel4	5:4	ADC channel 4 divider setting			0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4				

BITFIELD	BITS	DESCRIPTION	DECODE
ch_sel4	3:0	ADC Input Select for ADC Output Register set 4	0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved

[ADC\\_LIMIT5\\_0 \(0x528\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chLoLimit_I5[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION					DECODE	
chLoLimit_I5	7:0	ADC Output Register set 5 - low limit threshold value, low bits					0xXX: Low limit low byte	

[ADC\\_LIMIT5\\_1 \(0x529\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chHiLimit_I5[3:0]							
<b>Reset</b>	0xF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION					DECODE	
chHiLimit_I5	7:4	ADC Output Register set 5 - high limit threshold value, low bits					0bX: High limit low nibble	
chLoLimit_h5	1:0	ADC Output Register set 5 - low limit threshold value, high bits					0bXX: Low limit high bits	

[ADC\\_LIMIT5\\_2 \(0x52A\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h5[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
chHiLimit_h5	5:0	ADC Output Register set 5 - high limit threshold value, high bits	0bXXXXXX: High limit high bits

#### ADC\_LIMIT5\_3 (0x52B)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	div_sel5[1:0]		ch_sel5[3:0]			
<b>Reset</b>	—	—	0b00		0x3			
<b>Access Type</b>	—	—	Write, Read		Write, Read			

BITFIELD	BITS	DESCRIPTION	DECODE
div_sel5	5:4	ADC channel 5 divider setting	0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4
ch_sel5	3:0	ADC Input Select for ADC Output Register set 5	0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved

#### ADC\_LIMIT6\_0 (0x52C)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>					chLoLimit_l6[7:0]				
<b>Reset</b>					0x00				
<b>Access Type</b>					Write, Read				
BITFIELD	BITS	DESCRIPTION				DECODE			
chLoLimit_l6	7:0	ADC Output Register set 6 - low limit threshold value, low bits				0xx: Low limit low byte			

#### ADC\_LIMIT6\_1 (0x52D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>			chHiLimit_l6[3:0]		—	—	chLoLimit_h6[1:0]	
<b>Reset</b>			0xF		—	—	0b00	
<b>Access Type</b>			Write, Read		—	—	Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
chHiLimit_l6	7:4	ADC Output Register set 6 - high limit threshold value, low bits	0bX: High limit low nibble
chLoLimit_h6	1:0	ADC Output Register set 6 - low limit threshold value, high bits	0bXX: Low limit high bits

### ADC\_LIMIT6\_2 (0x52E)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h6[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					
BITFIELD	BITS	DESCRIPTION					DECODE	
chHiLimit_h6	5:0	ADC Output Register set 5 - high limit threshold value, high bits					0bXXXXXX: High limit high bits	

### ADC\_LIMIT6\_3 (0x52F)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	div_sel6[1:0]					ch_sel6[3:0]
<b>Reset</b>	—	—	0b00					0x3
<b>Access Type</b>	—	—	Write, Read					Write, Read
BITFIELD	BITS	DESCRIPTION					DECODE	
div_sel6	5:4	ADC channel 6 divider setting					0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4	
ch_sel6	3:0	ADC Input Select for ADC Output Register set 6					0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved	

[ADC\\_LIMIT7\\_0 \(0x530\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chLoLimit_I7[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
chLoLimit_I7	7:0	ADC Output Register set 7 - low limit threshold value, low bits			0xXX: Low limit low byte			

[ADC\\_LIMIT7\\_1 \(0x531\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	chHiLimit_I7[3:0]							
<b>Reset</b>	0xF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
chHiLimit_I7	7:4	ADC Output Register set 7 - high limit threshold value, low bits			0bX: High limit low nibble			
chLoLimit_h7	1:0	ADC Output Register set 7- low limit threshold value, high bits			0bXX: Low limit high bits			

[ADC\\_LIMIT7\\_2 \(0x532\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	chHiLimit_h7[5:0]					
<b>Reset</b>	—	—	0b111111					
<b>Access Type</b>	—	—	Write, Read					
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>			
chHiLimit_h7	5:0	ADC Output Register set 7 - high limit threshold value, high bits			0bXXXXXX: High limit high bits			

[ADC\\_LIMIT7\\_3 \(0x533\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	div_sel7[1:0]			ch_sel7[3:0]			
<b>Reset</b>	—	—	0b00			0x3			
<b>Access Type</b>	—	—	Write, Read			Write, Read			
BITFIELD	BITS	<b>DESCRIPTION</b>			<b>DECODE</b>				
div_sel7	5:4	ADC channel 7 divider setting			0b00: Divide by 1 0b01: Divide by 2 0b10: Divide by 3 0b11: Divide by 4				

BITFIELD	BITS	DESCRIPTION	DECODE
ch_sel7	3:0	ADC Input Select for ADC Output Register set 7	0x0: ADC0 (See adc_pin_en[0] bit) 0x1: ADC1 (See adc_pin_en[1] bit) 0x2: ADC2 (See adc_pin_en[2] bit) 0x3: Reserved 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: VDDIO/4 0x9: VDD18/2 0xA: CAP_VDD/2 0xB: TMON (Valid only in Round Robin mode) 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: Reserved

[ADC\\_RR\\_CTRL0 \(0x534\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	—	—	adc_rr_run
<b>Reset</b>	—	—	—	—	—	—	—	0b0
<b>Access Type</b>	—	—	—	—	—	—	—	Write, Read
BITFIELD	BITS	DESCRIPTION						DECODE
adc_rr_run	0	Enable ADC round robin operation						0b0: A/D conversions must be manually triggered using register bit cpu_adc_start 0b1: ADC runs in round robin mode, periodically sampling all enabled inputs

[ADC\\_RR\\_CTRL2 \(0x536\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	adc_rr_sleep_l[7:0]						—	—
<b>Reset</b>	0x00						—	—
<b>Access Type</b>	Write, Read						—	—
BITFIELD	BITS	DESCRIPTION						DECODE
adc_rr_sleep_l	7:0	Low byte of number of ADC conversion cycles the state machine sleeps between cycles.						0xXX: Low byte of number of ADC conversion cycles

[ADC\\_RR\\_CTRL3 \(0x537\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	adc_rr_sleep_h[7:0]						—	—
<b>Reset</b>	0x00						—	—
<b>Access Type</b>	Write, Read						—	—

BITFIELD	BITS	DESCRIPTION	DECODE
adc_rr_sleep_h	7:0	High byte of number of ADC conversion cycles the state machine sleeps between cycles.	0xXX: High byte of number of ADC conversion cycles

ADC\_CTRL\_4 (0x53E)

BIT	7	6	5	4	3	2	1	0
Field	RSVD	RSVD[3:0]					adc_pin_en[2:0]	
Reset	0b0	0x0					0b000	
Access Type							Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
adc_pin_en	2:0	Connect selected MFP pins to ADC input mux	0bXX0: Disable connection of MFP3 pin to ADC input mux 0bXX1: Enable connection of MFP3 pin to ADC input mux 0bX0X: Disable connection of MFP5 pin to ADC input mux 0bX1X: Enable connection of MFP5 pin to ADC input mux 0b0XX: Disable connection of MFP6 pin to ADC input mux 0b1XX: Enable connection of MFP6 pin to ADC input mux

UART\_PT\_0 (0x548)\*

BIT	7	6	5	4	3	2	1	0
Field	BITLEN_PT_1_L[7:0]							
Reset	0xDC							
Access Type	Write, Read							
BITFIELD	BITS	DESCRIPTION	DECODE					
BITLEN_PT_1_L	7:0	Low byte of custom (manually configured) 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.	0xXX: Low byte of custom UART bit length for pass-through UART Channel 1					

UART\_PT\_1 (0x549)\*

BIT	7	6	5	4	3	2	1	0
Field	—	—	BITLEN_PT_1_H[5:0]					
Reset	—	—	0b000101					
Access Type	—	—	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
BITLEN_PT_1_H	5:0	High byte of custom (manually configured) 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.	0XX: High byte of custom UART bit length for pass-through UART Channel 1

UART\_PT\_2 (0x54A)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	BITLEN_PT_2_L[7:0]							
<b>Reset</b>	0xDC							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
BITLEN_PT_2_L	7:0	Low byte of custom (manually configured) 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.	0XX: Low byte of custom UART bit length for pass-through UART Channel 2

UART\_PT\_3 (0x54B)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	BITLEN_PT_2_H[5:0]							
<b>Reset</b>	0b000101							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
BITLEN_PT_2_H	5:0	High byte of custom (manually configured) 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.	0XX: High byte of custom UART bit length for pass-through UART Channel 2

I2C\_PT\_4 (0x550)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SRC_A_1[6:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION
SRC_A_1	7:1	I <sup>2</sup> C address translator source A for pass-through Channel 1. When an I <sup>2</sup> C transaction across the GMSL link has a device address matching I <sup>2</sup> C SRC_A_1, the device address as seen on the remote side is replaced by the device address in I <sup>2</sup> C DST_A_1.

I2C\_PT\_5 (0x551)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DST_A_1[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—
BITFIELD	BITS		DESCRIPTION					
DST_A_1	7:1		I <sup>2</sup> C address translator destination A for pass-through Channel 1. See the description of I <sup>2</sup> C SRC_A_1.					

I2C\_PT\_6 (0x552)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SRC_B_1[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—
BITFIELD	BITS		DESCRIPTION					
SRC_B_1	7:1		I <sup>2</sup> C address translator source B for pass-through Channel 1. When an I <sup>2</sup> C transaction across the GMSL link has a device address matching I <sup>2</sup> C SRC_B_1, the device address as seen on the remote side is replaced by the device address in I <sup>2</sup> C DST_B_1.					

I2C\_PT\_7 (0x553)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DST_B_1[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—
BITFIELD	BITS		DESCRIPTION					
DST_B_1	7:1		I <sup>2</sup> C address translator destination B for pass-through Channel 1. See the description of I <sup>2</sup> C SRC_B_1.					

I2C\_PT\_8 (0x554)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SRC_A_2[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—
BITFIELD	BITS		DESCRIPTION					

BITFIELD	BITS	DESCRIPTION
SRC_A_2	7:1	<p>I<sup>2</sup>C address translator source A for pass-through Channel 2.</p> <p>When an I<sup>2</sup>C transaction across the GMSL link has a device address matching I<sup>2</sup>C SRC_A_2, the device address as seen on the remote side is replaced by the device address in I<sup>2</sup>C DST_A_2.</p>

**I<sup>2</sup>C\_PT\_9 (0x555)**

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DST_A_2[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—

BITFIELD	BITS	DESCRIPTION
DST_A_2	7:1	<p>I<sup>2</sup>C address translator destination A for pass-through Channel 2.</p> <p>See the description of I<sup>2</sup>C SRC_A_2.</p>

**I<sup>2</sup>C\_PT\_10 (0x556)**

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SRC_B_2[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—

BITFIELD	BITS	DESCRIPTION
SRC_B_2	7:1	<p>I<sup>2</sup>C address translator source B for pass-through Channel 2.</p> <p>When an I<sup>2</sup>C transaction across the GMSL link has a device address matching I<sup>2</sup>C SRC_B_2, the device address as seen on the remote side is replaced by the device address in I<sup>2</sup>C DST_B_2.</p>

**I<sup>2</sup>C\_PT\_11 (0x557)**

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DST_B_2[6:0]							—
<b>Reset</b>	0x00							—
<b>Access Type</b>	Write, Read							—

BITFIELD	BITS	DESCRIPTION
DST_B_2	7:1	<p>I<sup>2</sup>C address translator destination B for pass-through Channel 2.</p> <p>See the description of I<sup>2</sup>C SRC_B_2.</p>

[HS\\_VS\\_Z \(0x55F\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	DE_DET_Z	VS_DET_Z	HS_DET_Z	–	–	VS_POL_Z	HS_POL_Z
<b>Reset</b>	–	0b0	0b0	0b0	–	–	0b0	0b0
<b>Access Type</b>	–	Read Only	Read Only	Read Only	–	–	Read Only	Read Only
BITFIELD	BITS	DESCRIPTION					DECODE	
DE_DET_Z	6	DE activity is detected on video pipeline Z					0x0: DE is not detected 0x1: DE is detected	
VS_DET_Z	5	VS activity is detected on video pipeline Z					0x0: VS is not detected 0x1: VS is detected	
HS_DET_Z	4	HS activity is detected on video pipeline Z					0x0: HS is not detected 0x1: HS is detected	
VS_POL_Z	1	Detected VS polarity on video pipeline Z					0x0: Active low 0x1: Active high	
HS_POL_Z	0	Detected HS polarity on video pipeline Z					0x0: Active low 0x1: Active high	

[UNLOCK\\_KEY \(0x56E\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	UNLOCK_KEY[7:0]							
<b>Reset</b>	0xBB							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION					DECODE	
UNLOCK_KEY	7:0	Register must be at unlock value to enable write access to port control bit fields. DIS_LOCAL_CC, IIC_1_EN, IIC_2_EN, UART_1_EN, UART_2_EN. Defaults to unlocked.					0xBB: Unlock write access Others: Lock write access	

[PIO\\_SLEW\\_0 \(0x56F\)\\*](#)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	–	–	PIO02_SLEW[1:0]			PIO01_SLEW[1:0]		PIO00_SLEW[1:0]		
<b>Reset</b>	–	–	0b11			0b11		0b10		
<b>Access Type</b>	–	–	Write, Read			Write, Read		Write, Read		
BITFIELD	BITS		DESCRIPTION							
PIO02_SLEW		5:4	Slew rate setting for MFP2 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.							
PIO01_SLEW		3:2	Slew rate setting for MFP1 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.							
PIO00_SLEW		1:0	Slew rate setting for MFP0 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.							

[PIO\\_SLEW\\_1 \(0x570\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	PIO06_SLEW[1:0]		PIO05_SLEW[1:0]		—	—
<b>Reset</b>	—	—	0b11		0b11		—	—
<b>Access Type</b>	—	—	Write, Read		Write, Read		—	—
BITFIELD	BITS		DESCRIPTION					
PIO06_SLEW	5:4		Slew rate setting for MFP4 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.					
PIO05_SLEW	3:2		Slew rate setting for MFP3 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.					

[PIO\\_SLEW\\_2 \(0x571\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PIO011_SLEW[1:0]		PIO010_SLEW[1:0]		RSVD[1:0]		—	—
<b>Reset</b>	0b11		0b11		0b11		—	—
<b>Access Type</b>	Write, Read		Write, Read				—	—
BITFIELD	BITS		DESCRIPTION					
PIO011_SLEW	7:6		Slew rate setting for MFP8 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.					
PIO010_SLEW	5:4		Slew rate setting for MFP7 pin. 00 value is the fastest rise and fall time, and 11 value is the slowest.					

[EXT4 \(0x584\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_fs_cnt_l[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS		DESCRIPTION					
ctrl1_fs_cnt_l	7:0		Frame start counter value of the virtual channel selected by FS_VC_SEL (low byte)					

[EXT5 \(0x585\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_fs_cnt_h[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS		DESCRIPTION					
ctrl1_fs_cnt_h	7:0		Frame start counter value of the virtual channel selected by FS_VC_SEL (high byte)					

EXT6 (0x586)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_fe_cnt_l[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS		DESCRIPTION					
ctrl1_fe_cnt_l	7:0		Frame end counter value of the virtual channel selected by FS_VC_SEL (low byte)					

EXT7 (0x587)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ctrl1_fe_cnt_h[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS		DESCRIPTION					
ctrl1_fe_cnt_h	7:0		Frame end counter value of the virtual channel selected by FS_VC_SEL (high byte)					

EXT8 (0x588)

BIT	7	6	5	4	3	2	1	0				
<b>Field</b>	—	—	—	—	ctrl1_fs_vc_sel[3:0]							
<b>Reset</b>	—	—	—	—	0x0							
<b>Access Type</b>	—	—	—	—	Write, Read							
BITFIELD	BITS		DESCRIPTION									
ctrl1_fs_vc_sel	3:0		Selected virtual channel for frame start/end count monitoring									

SPI\_CC\_WR (0x1300)

SPI data to write over the GMSL link. Use this address space to send write data across the GMSL link.

SPI\_CC\_RD (0x1380)

SPI data received over the GMSL link. Use this address space to read data sent across the GMSL link.

RLMS3 (0x1403)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	AdaptEn	AdpHaltLO S	AdpStartTrn	AdpTimHld	GearShEn	—	GearSh[1:0]	
<b>Reset</b>	0b0	0b0	0b0	0b0	0b1	—	0b10	
<b>Access Type</b>	Write, Read	—	Write, Read					

BITFIELD	BITS	DESCRIPTION	DECODE
AdaptEn	7	Adapt process enable	
AdpHaltLOS	6	Halt adapt process on loss-of-signal	0: Ignore LOS for adapt process control 1: Halt adapt process on LOS, restart on signal detect (assuming AdaptEn is active)
AdpStartTrn	5	Start adapt process on training active	0: Ignore training active for adapt process control 1: Start adapt process on training active (assuming AdaptEn is active)
AdpTimHld	4	Hold adapt timers even if adapt process is idle	0: Adapt timers (gear shift, CTF delay, DFE delay) operate normally and are cleared when adapt process goes to idle (halt) 1: Adapt timers hold their values even if adapt process goes to idle (halt)
GearShEn	3	Gear shift enable	
GearSh	1:0	Gear shift adapt gain shift select	

RLMS4 (0x1404)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	EOM_CHK_AMOUNT[3:0]				EOM_CHK_THR[1:0]		EOM_PER_MODE	EOM_EN
<b>Reset</b>	0x4				0b10		0b1	0b1
<b>Access Type</b>	Write, Read				Write, Read		Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
EOM_CHK_AMOUNT	7:4	A factor (N) used to select the order of number of observations in each eye-opening monitor window. N is used in the equation: Observations = $6.29 \times 10^4 (N + 2)$	0xX: N factor
EOM_CHK_THR	3:2	Eye-opening monitor number of error bits to allow in a measurement window	0b00: Allow no errors 0b01: Allow 1 error 0b10: Allow 2 errors 0b11: Allow 3 errors
EOM_PER_MODE	1	Eye-opening monitor periodic mode enable	0b0: Eye opening monitor periodic mode disabled 0b1: Eye opening monitor periodic mode enabled
EOM_EN	0	Eye-opening monitor enable	0b0: Eye opening monitor disabled 0b1: Eye opening monitor enabled

RLMS5 (0x1405)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	EOM_MAN_TRG_REQ	EOM_MIN_THR[6:0]						
<b>Reset</b>	0b0	0b0010000						
<b>Access Type</b>	Write Only	Write, Read						

BITFIELD	BITS	DESCRIPTION	DECODE
EOM_MAN_TRG_REQ	7	Eye-opening monitor manual trigger. For use when periodic mode is disabled.	0b0: No action 0b1: EOM manual trigger request
EOM_MIN_THR	6:0	The EOM minimum threshold as defined by the equation: % eye opening = EOM_MIN_THR/64. If measured, eye opening falls below this threshold, ERRB is triggered if enabled by EOM_ERR_FLAG_*. If the value is zero, the EOM is disabled.	0bXXXXXXXX: EOM minimum threshold factor

RLMS6 (0x1406)\*

BIT	7	6	5	4	3	2	1	0
Field	EOM_PV_MODE	RSVD[6:0]						
Reset	0b1	0b0000000						
Access Type	Write, Read							
BITFIELD	BITS	DESCRIPTION			DECODE			
EOM_PV_MODE	7	Eye-opening is measured vertically or horizontally			0b0: Vertical opening mode 0b1: Horizontal opening mode			

RLMS7 (0x1407)

BIT	7	6	5	4	3	2	1	0
Field	EOM_DONE	EOM[6:0]						
Reset	0b0	0b0000000						
Access Type	Read Only	Read Only						
BITFIELD	BITS	DESCRIPTION			DECODE			
EOM_DONE	7	Eye-opening monitor measurement done			0b0: EOM not complete 0b1: EOM complete			
EOM	6:0	Last completed EOM observation For horizontal eye-opening measurement, eye-opening is 2 x EOM/127 UI For vertical eye-opening measurement, eye-opening is EOM/127 UI			0bXXXXXXXX: EOM measurement result			

RLMS17 (0x1417)

BIT	7	6	5	4	3	2	1	0
Field	DFE5En	DFE4En	DFE3En	DFE2En	DFE1En	BSTEnOv	BSTEn	AGCEn
Reset	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read
BITFIELD	BITS		DESCRIPTION					
DFE5En	7		DFE5 coefficient adapt enable					
DFE4En	6		DFE4 coefficient adapt enable					

BITFIELD	BITS	DESCRIPTION
DFE3En	5	DFE3 coefficient adapt enable
DFE2En	4	DFE2 coefficient adapt enable
DFE1En	3	DFE1 coefficient adapt enable
BSTEnOv	2	When 1, BSTEn from is set from registers
BSTEn	1	Frequency boost adapt enable (Disabled by default for Slow receiver)
AGCEn	0	AGC adapt enable

RLMS1C (0x141C)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	AGCMuL[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	DESCRIPTION							
AGCMuL	AGC adapt gain LSB							

RLMS1D (0x141D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	AGCMuH[5:0]					
<b>Reset</b>	—	—	0b000010					
<b>Access Type</b>	—	—	Write, Read					
BITFIELD	DESCRIPTION							
AGCMuH	AGC adapt gain MSB							

RLMS1F (0x141F)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	AGCInit[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	DESCRIPTION							
AGCInit	AGC initial value							

RLMS32 (0x1432)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	OSNMode	RSVD	RSVD[5:0]					
<b>Reset</b>	0b0	0b1	0b111111					
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
OSNMode	7	GMSL2 OSN mode	0b0: Duty cycle 0b1: Average Ek

RLMS3A (0x143A)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	EyeMonValCntL[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	DESCRIPTION							
EyeMonValCntL	7:0						Eye monitor valid (hit) count (read-only)	

RLMS3B (0x143B)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	EyeMonValCntH[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	DESCRIPTION							
EyeMonValCntH	7:0						Eye monitor valid (hit) count (read-only)	

RLMS64 (0x1464)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD[3:0]							
<b>Reset</b>	0x9							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
TxSSCMode	1:0	Tx spread-spectrum mode						00: Manual phase mode (SSC disabled) 01: Reserved 10: Reserved 11: SSC enabled

RLMS70 (0x1470)\*

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	TxSSCFrqCtrl[6:0]							
<b>Reset</b>	—	0b0000001							
<b>Access Type</b>	—	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
TxSSCFrqCtr_I	6:0	Tx SSC modulation frequency deviation control (pp)	0x07: SSC 268 PPM 0x06: SSC 580 PPM 0x03: SSC 970 PPM 0x01: SSC 1750 PPM 0x01: SSC 2530 PPM others: Reserved

[RLMS71 \(0x1471\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	TxSSCCenSprSt[5:0]							TxSSCEn
<b>Reset</b>	—	0b000001							0b0
<b>Access Type</b>	—	Write, Read			Write, Read				
BITFIELD	BITS	DESCRIPTION			DECODE				
TxSSCCenSprSt	6:1	Tx SSC center spread starting phase			0x02: SSC 268 PPM 0x02: SSC 580 PPM 0x02: SSC 970 PPM 0x02: SSC 1750 PPM 0x02: SSC 2530 PPM others: Reserved				
TxSSCEn	0	Tx spread spectrum enable			0b0: Tx spread spectrum disabled 0b1: Tx spread spectrum enabled				

[RLMS72 \(0x1472\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	TxSSCPreSclL[7:0]								
<b>Reset</b>	0xCF								
<b>Access Type</b>	Write, Read								
BITFIELD	BITS	DESCRIPTION			DECODE				
TxSSCPreSclL	7:0	Tx SSC frequency prescaler bits 7:0. Decode values are for bits 7:0, concatenate with TXSSCPreSclH for final value.			0xC9: SSC 268 PPM 0xAB: SSC 580 PPM 0xAB: SSC 970 PPM 0xF9: SSC 1750 PPM 0xAB: SSC 2530 PPM others: Reserved				

[RLMS73 \(0x1473\)\\*](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	—	—	—	—	TxSSCPreSclH[2:0]			
<b>Reset</b>	—	—	—	—	—	0b000			
<b>Access Type</b>	—	—	—	—	—	Write, Read			

BITFIELD	BITS	DESCRIPTION	DECODE
TxSSCPreScIH	2:0	Tx SSC frequency prescaler bits 10:8. Decode values are for bits 10:8, concatenate with TXSSCPreScL for final value.	0x02: SSC 268 PPM 0x00: SSC 580 PPM 0x00: SSC 970 PPM 0x00: SSC 1750 PPM 0x00: SSC 2530 PPM others: Reserved

**RLMS74 (0x1474)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TxSSCPhL[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE
TxSSCPhL	7:0	Tx SSC phase accumulator increment bits 7:0. Decode values are for bits 7:0, concatenate with TXSSCPhH for final value.						0xF9: SSC 268 PPM 0x63: SSC 580 PPM 0x63: SSC 970 PPM 0x2C: SSC 1750 PPM 0x63: SSC 2530 PPM others: Reserved

**RLMS75 (0x1475)\***

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	TxSSCPhH[6:0]							
<b>Reset</b>	—	0b0000000							
<b>Access Type</b>	—	Write, Read							
BITFIELD	BITS	DESCRIPTION						DECODE	
TxSSCPhH	6:0	Tx SSC phase accumulator increment bits 14:8. Decode values are for bits 14:8, concatenate with TXSSCPhL for final value.						0x01: SSC 268 PPM 0x07: SSC 580 PPM 0x07: SSC 970 PPM 0x05: SSC 1750 PPM 0x07: SSC 2530 PPM others: Reserved	

**RLMS76 (0x1476)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	—	TxSSCPhQuad[1:0]	
<b>Reset</b>	—	—	—	—	—	—	0b00	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
TxSSCPhQuad	1:0	Tx SSC phase starting phase quadrant	0x00: SSC 268 PPM 0x00: SSC 580 PPM 0x00: SSC 970 PPM 0x00: SSC 1750 PPM 0x00: SSC 2530 PPM others: Reserved

RLMS84 (0x1484)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TuneAmp_ManL	TuneOff_Man[6:0]						
<b>Reset</b>	0b0	0b0000000						
<b>Access Type</b>	Write, Read	Write, Read						
BITFIELD	BITS	DESCRIPTION						
TuneAmp_ManL	7	Manual override for amplitude calibration value LSB						
TuneOff_Man	6:0	Manual override for offset calibration value						

RLMS85 (0x1485)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TuneAmp_ManH[7:0]							
<b>Reset</b>	0x3B							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						
TuneAmp_ManH	7:0	Manual override for amplitude calibration value MSB						

RLMS95 (0x1495)

BIT	7	6	5	4	3	2	1	0						
<b>Field</b>	TxAmpManEn	ADC Sar Met hod	TxAmpMan[5:0]											
<b>Reset</b>	0b0	0b1	0b101001											
<b>Access Type</b>	Write, Read	Write, Read	Write, Read											
BITFIELD	BITS	DESCRIPTION												
TxAmpManEn	7	TX amplitude manual override												
ADC Sar Method	6	0=6 bit SAR, 1=7 bit SAR												
TxAmpMan	5:0	Tx amplitude												

RLMSA8 (0x14A8)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	FW_PHY_C TRL	FW_PHY_P U_TX	FW_PHY_R STB	FW_HSK_P HASE	FW_HSK_T YPE	FW_HSK_T X_EN	FW_HSK_R X_EN	FW_CALIB RATING
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
FW_PHY_CTRL	7	PHY controller firmware mode enable. Other FW_* bits only take effect when this is set to 1.	0b0: Disabled 0b1: Enabled
FW_PHY_PU_TX	6	Override PHY controller output	0b0: Disabled 0b1: Enabled
FW_PHY_RSTB	5	Override PHY controller output	0b0: Reset asserted 0b1: Reset not asserted
FW_HSK_PHASE	4	Override PHY controller output	
FW_HSK_TYPE	3	Override PHY controller output	
FW_HSK_TX_EN	2	Override PHY controller output	
FW_HSK_RX_EN	1	Override PHY controller output	
FW_CALIBRATING	0	Override PHY controller output	

RLMSA9 (0x14A9)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	FW_REPCAL_RSTB	FW_START_CAP_CAL	FW_TXD_SQUELCH	FW_TXD_EN	FW_RXD_EN	FW_AEQ_EN	FW_WBL_EN	FW_HANDSHAKED
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b0
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
FW_REPCAL_RSTB	7	Override PHY controller output	0b0: Reset asserted 0b1: Reset not asserted
FW_START_CAP_CAL	6	Override PHY controller output	
FW_TXD_SQUELCH	5	Override PHY controller output	0b0: Disabled 0b1: Enabled
FW_TXD_EN	4	Override PHY controller output	0b0: Disabled 0b1: Enabled
FW_RXD_EN	3	Override PHY controller output	0b0: Disabled 0b1: Enabled
FW_AEQ_EN	2	Override PHY controller output	

BITFIELD	BITS	DESCRIPTION				DECODE			
FW_WBL_EN	1	Override PHY controller output							
FW_HANDSHAKED	0	Override PHY controller output							

[RLMSAA \(0x14AA\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	RSVD	ROR_CLK_DET	RSVD	RSVD	RSVD	RSVD	RSVD
<b>Reset</b>	0b1	0b0	0b0	0b1	0b0	0b0	0b0	0b0
<b>Access Type</b>			Read Only					
BITFIELD	BITS	DESCRIPTION						DECODE
ROR_CLK_DET	5	In SER, indicates ROR clock is detected.						0x0: ROR clock has not been detected (or ROR mode is not enabled) 0x1: ROR clock has been detected

[RLMSBA \(0x14BA\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TwoDAmpMin[7:0]							
<b>Reset</b>	0x30							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION						
TwoDAmpMin	7:0	Minimum allowed Amp value for 2D calibration algorithm						

[RLMSC8 \(0x14C8\)](#)

BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	—	VmITxAmpl_reg[6:0]							
<b>Reset</b>	—	0b1111111							
<b>Access Type</b>	—	Write, Read							
BITFIELD	BITS	DESCRIPTION							
VmITxAmpl_reg	6:0	value to use if manual control							

[RLMSC9 \(0x14C9\)](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	mcursor_reg[3:0]			
<b>Reset</b>	—	—	—	—	0x5			
<b>Access Type</b>	—	—	—	—	Write, Read			

BITFIELD	BITS	DESCRIPTION
mcursor_reg	3:0	value to use if manual control

RLMSCA (0x14CA)

BIT	7	6	5	4	3	2	1	0			
<b>Field</b>	—	—	—	—	rcursor_reg[3:0]						
<b>Reset</b>	—	—	—	—	0x7						
<b>Access Type</b>	—	—	—	—	Write, Read						

BITFIELD	BITS	DESCRIPTION
rcursor_reg	3:0	value to use if manual control

RLMSCE (0x14CE)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	RSVD	RSVD	enminus_re g	enminus_m an	RSVD	RSVD	enffe
<b>Reset</b>	0b0	0b0	0b0	0b0	0b0	0b0	0b0	0b1
<b>Access Type</b>				Write, Read	Write, Read			Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
enminus_reg	4	value to use if manual control enabled with enminus_man register bit	
enminus_ma n	3	enminus manual control	0x0: Automatic 0x1: Manual register control
enffe	0	ffe enable	0x0: Disabled 0x1: Enabled

DPLL\_0 (0x1A00)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	RSVD	config_soft_ rst_n						
<b>Reset</b>	0b1	0b1	0b1	0b1	0b0	0b1	0b0	0b1
<b>Access Type</b>								Write, Read

BITFIELD	BITS	DESCRIPTION
config_soft_rst_n	0	Setting this to 1 resets the PLL functional registers. PLL configuration registers are not reset. This bit must be set back to 0 for the PLL to be functional.

DPLL\_3 (0x1A03)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	config_sel_clock_out_use_external	RSVD	RSVD	config_use_internal_divider_values	config_force_enable_ss	config_spread_bit_ratio[2:0]		
<b>Reset</b>	0b1	0b0	0b0	0b0	0b0	0b010		
<b>Access Type</b>	Write, Read			Write, Read	Write, Read	Write, Read		
BITFIELD	BITS	DESCRIPTION				DECODE		
config_sel_clock_out_use_external	7	When 1, config_sel_clock_out is used to select output clock. Otherwise, internal registers are used.						
config_use_internal_divider_values	4	Enable all DPLL divider values to come from internal controls				0x0: Do not use internal controls for divider values 0x1: Use internal controls for divider values		
config_force_enable_ss	3	Enables the output of the triangle wave generator (overrides en_ss to force triangle_wave on)						
config_spread_bit_ratio	2:0	Controls the magnitude of the triangle wave input to the divider dsm as a percentage of the nominal divider value. If config registers are reset, the spread_bit_ratio value does not propagate to the triangle wave without rewriting to it. Likewise, if triangle wave module is reset, rewrite to spread_bit_ratio to set it back to desired value.						

DPLL\_7 (0x1A07)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	config_div_fb_L	config_div_in[4:0]						RSVD[1:0]
<b>Reset</b>	0b0	0b00001						0b00
<b>Access Type</b>	Write, Read	Write, Read						
BITFIELD	BITS	DESCRIPTION						
config_div_fb_L	7	Sets the DPLL feedback divider value (bit 0) when config_use_internal_divider_values = 1						
config_div_in	6:2	Sets the divide value of the input divider connected to i_clk_in, the main PLL clock input						

DPLL\_8 (0x1A08)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	config_div_fb_H[7:0]							
<b>Reset</b>	0x14							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION
config_div_fb_H	7:0	Sets the DPLL feedback divider value (bits 8:1) when config_use_internal_divider_values = 1

DPLL\_9 (0x1A09)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	config_div_out_L[4:0]						config_div_fb_exp[2:0]	
<b>Reset</b>	0b01000						0b000	
<b>Access Type</b>	Write, Read						Write, Read	

BITFIELD	BITS	DESCRIPTION
config_div_out_L	7:3	Sets the DPLL output divider value (bits 4:0) when config_use_internal_divider_values = 1
config_div_fb_exp	2:0	Sets the feedback exponential divider value when config_use_internal_divider_values = 1

DPLL\_10 (0x1A0A)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	config_allow_coarse_change	config_div_out_exp[2:0]			config_div_out_H[3:0]			
<b>Reset</b>	0b1	0b000			0x1			
<b>Access Type</b>	Write, Read	Write, Read			Write, Read			

BITFIELD	BITS	DESCRIPTION
config_allow_coarse_change	7	When set to 1, the coarse tuning DAC is allowed to move to correct for large changes in the integral path. When set to 0, it does not.
config_div_out_exp	6:4	Sets the output exponential divider value when config_use_internal_divider_values = 1.
config_div_out_H	3:0	Sets the DPLL output divider value (bits 8:5) when config_use_internal_divider_values = 1.

EFUSE80 (0x1C50)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	SERIAL_NUMBER_0[7:0]									
<b>Reset</b>	0x00									
<b>Access Type</b>	Read Only									
BITFIELD	BITS	DESCRIPTION								
SERIAL_NUMBER_0	7:0	Serial Number. Can only be read through the deserializer across the GMSL link.								

[EFUSE81 \(0x1C51\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_1[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_1	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE82 \(0x1C52\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_2[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_2	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE83 \(0x1C53\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_3[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_3	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE84 \(0x1C54\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_4[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_4	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE85 \(0x1C55\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_5[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_5	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE86 \(0x1C56\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_6[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_6	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE87 \(0x1C57\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_7[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_7	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE88 \(0x1C58\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_8[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_8	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE89 \(0x1C59\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_9[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_9	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE90 \(0x1C5A\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_10[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_10	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE91 \(0x1C5B\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_11[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_11	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE92 \(0x1C5C\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_12[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_12	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE93 \(0x1C5D\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_13[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_13	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE94 \(0x1C5E\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_14[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_14	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE95 \(0x1C5F\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_15[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_15	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE96 \(0x1C60\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_16[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_16	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE97 \(0x1C61\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_17[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_17	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE98 \(0x1C62\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_18[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_18	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE99 \(0x1C63\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_19[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_19	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE100 \(0x1C64\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_20[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_20	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE101 \(0x1C65\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_21[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_21	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE102 \(0x1C66\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_22[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_22	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[EFUSE103 \(0x1C67\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SERIAL_NUMBER_23[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS		DESCRIPTION					
SERIAL_NUMBER_23	7:0		Serial Number. Can only be read through the deserializer across the GMSL link.					

[REGCRC0 \(0x1D00\)\\*](#)

BIT	7	6	5	4	3	2	1	0
Field	—	—	LIMIT_REG_RANGE	GEN_ROLLING_CRC	I2C_WR_COMPUTE	PERIODIC_COMPUTE	CHECK_CRC	RESET_CRC
Reset	—	—	0b0	0b0	0b0	0b0	0b0	0b0
Access Type	—	—	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write Clears All, Read
BITFIELD	BITS		DESCRIPTION				DECODE	
LIMIT_REG_RANGE	5		Limit the min. and max. range of registers calculated					
GEN_ROLLING_CRC	4		Calculate register CRC using additional 2-bit counter, so CRC value cycles every four invocations.				0b0: Do not use additional counter 0b1: Use additional counter	

BITFIELD	BITS	DESCRIPTION	DECODE
I2C_WR_CO MPUTE	3	Compute register CRC after every I <sup>2</sup> C register write.	0b0: Do not compute after every register write 0b1: Compute after every register write
PERIODIC_ COMPUTE	2	Check register CRC on periodic basis, based on CRC_PERIOD value.	0b0: Do not check periodically 0b1: Check periodically
CHECK_CR C	1	Upon calculation of register CRC, compare with previous calculation, except on first time through. On miscompare, issue ERRB.	0b0: Do not compare after calculation 0b1: Compare after calculation
RESET_CRC	0	Reset CRC value to 16'FFFF.	

### REGCRC1 (0x1D01)\*

BIT	7	6	5	4	3	2	1	0
Field	CRC_PERIOD[7:0]							
Reset	0x00							
Access Type	Write, Read							
BITFIELD	BITS			DESCRIPTION				
CRC_PERIOD	7:0			Period for register CRC recomputation. This allows for varying the CRC computation time. Period = (value + 1) * 2ms  0000_0000 - 2ms 0000_0001 - 4ms ...				

### REGCRC2 (0x1D02)

BIT	7	6	5	4	3	2	1	0
Field	REGCRC LSB[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS			DESCRIPTION				
REGCRC LSB	7:0			Calculated register CRC value (LSB)				

### REGCRC3 (0x1D03)

BIT	7	6	5	4	3	2	1	0
Field	REGCRC MSB[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS			DESCRIPTION				
REGCRC MSB	7:0			Calculated register CRC value (MSB)				

I2C\_UART\_CRC0 (0x1D08)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	–	–	–	–	–	–	–	RESET_MS_GCNTR
<b>Reset</b>	–	–	–	–	–	–	–	0b0
<b>Access Type</b>	–	–	–	–	–	–	–	Write Clears All, Read
BITFIELD	BITS	<b>DESCRIPTION</b>					<b>DECODE</b>	
RESET_MS_GCNTR	0	Reset Message Counter Value to 0. Self-clearing. Note that the message counter value is not checked when writing to this register. See user guide for details.					0b0: Do not reset 0b1: Reset	

I2C\_UART\_CRC1 (0x1D09)\*

BIT	7	6	5	4	3	2	1	0			
<b>Field</b>	RSVD[2:0]			RSVD[2:0]			RESET_MS_GCNTR_ER_R_CNT	RESET_CRC_ERR_CNT			
<b>Reset</b>	0b000			0b000			0b0	0b0			
<b>Access Type</b>							Write Clears All, Read	Write Clears All, Read			
BITFIELD	BITS	<b>DESCRIPTION</b>					<b>DECODE</b>				
RESET_MS_GCNTR_ER_R_CNT	1	Reset message counter error count to 0. Self-clearing.					0b0: Do not reset 0b1: Reset				
RESET_CRC_ERR_CNT	0	Reset CRC Error Count to 0. Self-clearing.					0b0: Do not reset 0b1: Reset				

I2C\_UART\_CRC2 (0x1D0A)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	CRC_VAL[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS	<b>DESCRIPTION</b>						
CRC_VAL	7:0	Calculated CRC value for the last write transaction						

I2C\_UART\_CRC3 (0x1D0B)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MSGCNTR LSB[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							

BITFIELD	BITS	DESCRIPTION
MSGCTR_LSB	7:0	Bits 7:0 of current message counter value

I2C\_UART\_CRC4 (0x1D0C)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MSGCTR_MSB[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
BITFIELD	BITS		DESCRIPTION					
MSGCTR_MSB	7:0		Bits 15:8 of current message counter value					

FS\_INTR0 (0x1D12)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	I2C_UART_MSGCTR_ERR_OEN	I2C_UART_CRC_ERR_OEN	MEM_ECC_ERR2_OEN	MEM_ECC_ERR1_OEN	–	–	–	REG_CRC_ERR_OEN
<b>Reset</b>	0b1	0b1	0x1	0x0	–	–	–	0b0
<b>Access Type</b>	Write, Read	Write, Read	Write, Read	Write, Read	–	–	–	Write, Read
BITFIELD	BITS		DESCRIPTION			DECODE		
I2C_UART_MSGCTR_ERR_OEN	7		Enable reporting of I <sup>2</sup> C/UART message counter errors (I2C_UART_MSGCTR_ERR) at ERRB pin			0b0: Reporting disabled 0b1: Reporting enabled		
I2C_UART_CRC_ERR_OEN	6		Enable reporting of I <sup>2</sup> C/UART CRC errors (I2C_UART_CRC_ERR) at ERRB pin			0b0: Reporting disabled 0b1: Reporting enabled		
MEM_ECC_ERR2_OEN	5		Enable reporting of memory ECC 2-bit uncorrectable errors at ERRB pin			0b0: Reporting disabled 0b1: Reporting enabled		
MEM_ECC_ERR1_OEN	4		Enable reporting of memory ECC 1-bit correctable errors at ERRB pin			0b0: Reporting disabled 0b1: Reporting enabled		
REG_CRC_ERR_OEN	0		Enable reporting register CRC at ERRB pin.			0b0: Reporting disabled 0b1: Reporting enabled		

FS\_INTR1 (0x1D13)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	I2C_UART_MSGCTR_ERR_INT	I2C_UART_CRC_ERR_INT	MEM_ECC_ERR2_INT	MEM_ECC_ERR1_INT	–	–	–	REG_CRC_ERR_FLAG
<b>Reset</b>	0b0	0b0	0x0	0x0	–	–	–	0b0
<b>Access Type</b>	Read Only	Read Only	Read Only	Read Only	–	–	–	Read Only

BITFIELD	BITS	DESCRIPTION	DECODE
I2C_UART_MSGCTR_ERR_INT	7	I2C/UART message counter error flag. Asserted when two message counter bytes sent do not match expected count value. See user guide for details.	0b0: Flag not asserted 0b1: Flag asserted
I2C_UART_CRC_ERR_INT	6	I2C/UART CRC error flag. Asserted when CRC byte sent does not match calculated value. See user guide for details.	0b0: Flag not asserted 0b1: Flag asserted
MEM_ECC_ERR2_INT	5	Error flag for 2-bit uncorrectable memory ECC errors seen in any memories	0b0: Flag not asserted 0b1: Flag asserted
MEM_ECC_ERR1_INT	4	Error flag for 1-bit correctable memory ECC errors seen in any memories	0b0: Flag not asserted 0b1: Flag asserted
REG_CRC_ERR_FLAG	0	An error occurred on the register CRC calculation.	0b0: Flag not asserted 0b1: Flag asserted

MEM\_ECC0 (0x1D14)\*

BIT	7	6	5	4	3	2	1	0
Field	RSVD[2:0]					RSVD[2:0]	RESET_ME_M_ECC_ER_R2_CNT	RESET_ME_M_ECC_ER_R1_CNT
Reset	0x0					0x0	0b0	0b0
Access Type							Write Clears All, Read	Write Clears All, Read
BITFIELD	BITS	DESCRIPTION					DECODE	
RESET_ME_M_ECC_ER_R2_CNT	1	Reset memory ECC 2-bit error count to 0. Self-clearing.					0b0: Do not reset 0b1: Reset	
RESET_ME_M_ECC_ER_R1_CNT	0	Reset memory ECC 1-bit error count to 0. Self-clearing.					0b0: Do not reset 0b1: Reset	

MEM\_ECC1 (0x1D15)

BIT	7	6	5	4	3	2	1	0
Field	MEM_ECC_ERR1_CNT[7:0]							
Reset	0x00							
Access Type	Read Only							
BITFIELD	BITS	DESCRIPTION						
MEM_ECC_ERR1_CNT	7:0	Number of 1-bit correctable memory ECC errors seen						

MEM\_ECC2 (0x1D16)

BIT	7	6	5	4	3	2	1	0
Field	MEM_ECC_ERR2_CNT[7:0]							
Reset	0x00							
Access Type	Read Only							

BITFIELD	BITS	DESCRIPTION
MEM_ECC_ERR2_CNT	7:0	Number of 2-bit uncorrectable memory ECC errors seen

MEM\_ECC3 (0x1D17)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	—	MEM_ECC_ERR2_INJ	MEM_ECC_ERR1_INJ
<b>Reset</b>	—	—	—	—	—	—	0b0	0b0
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
MEM_ECC_ERR2_INJ	1	Inject memory ECC 2-bit errors. Injects error to memory instance given by MEM_ECC_ERR_INJ_SEL register.	0: Do not inject memory ECC errors 1: Inject memory ECC errors
MEM_ECC_ERR1_INJ	0	Inject memory ECC 1-bit errors. Injects error to memory instance given by MEM_ECC_ERR_INJ_SEL register.	0: Do not inject memory ECC errors 1: Inject memory ECC errors

MEM\_ECC4 (0x1D18)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MEM_ECC_ERR_INJ_SEL[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
MEM_ECC_ERR_INJ_SEL	7:0	Selects which memory instance to inject ECC errors to. Encoding is the same as MBIST memory selection.	0: All memories with ECC 3: ARQ GPIO 4: TBB 10: Video TX Z 11: Video TX Z 14: FrontTop 30: ARQ SPI others: none

REG\_POST0 (0x1D20)\*

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	POST_DON_E	POST_MBI_ST_PASSED	POST_LBIS_T_PASSED	—	—	RSVD	RSVD	RSVD
<b>Reset</b>	0b0	0b0	0b0	—	—	0b0	0b0	0b0
<b>Access Type</b>	Read Only	Read Only	Read Only	—	—			

BITFIELD	BITS	DESCRIPTION	DECODE
POST_DON_E	7	POST (power-on self test LBIST and/or MBIST) is run.	0b0: POST has not run 0b1: POST has run

BITFIELD	BITS	DESCRIPTION	DECODE
POST_MBIST_PASSED	6	MBIST passed during POST (power-on self test). Valid when POST_DONE is asserted.	0b0: MBIST has failed during POST run (or has not been enabled) 0b1: MBIST has passed during POST run
POST_LBIST_PASSED	5	LBIST passed during POST (power-on self test). Valid when POST_DONE is asserted.	0b0: LBIST has failed during POST run (or has not been enabled) 0b1: LBIST has passed during POST run

[REGADCBIST0 \(0x1D28\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
Field	RR_ACCURACY	RSVD	–	MUXVER_EN	–	RUN_ACCURACY	RSVD	RUN_TMON_CAL
Reset	0b0	0b0	–	0b0	–	0b0	0b0	0b0
Access Type	Write, Read		–	Write, Read	–	Write Clears All, Read		Write Clears All, Read

BITFIELD	BITS	DESCRIPTION	DECODE
RR_ACCURACY	7	Run ADC accuracy in round robin state fashion	0b0: Do not run 0b1: Run
MUXVER_EN	4	Enable MUX manual GPIO test	0b0: Disable 0b1: Enable
RUN_ACCURACY	2	Run ADC accuracy testing. Self-clearing.	0b0: Do not run 0b1: Run
RUN_TMON_CAL	0	Initiate temperature sensor measurement. Self-clearing.	0b0: Do not run 0b1: Run

[REGADCBIST3 \(0x1D31\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
Field	REFLIM[7:0]							
Reset	0x0F							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
REFLIM	7:0	ADC reference limit for testing code. Sets maximum error from expected ADC conversion code in LSBs before asserting interrupt flag.	0x00: Converted value has zero deviation 0x01: Converted value has up to 1 LSB of error 0x02: Converted value has up to 2 LSBs of error ... 0xFF: Converted value has up to 255 LSBs of error

[REGADCBIST4 \(0x1D32\)](#)<sup>\*</sup>

BIT	7	6	5	4	3	2	1	0
Field	REFLIMSCL1[7:0]							
Reset	0x0F							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
REFLIMSCL 1	7:0	Control the accuracy for 1/2 scale measurement. Sets maximum error from expected ADC conversion code in LSBs before asserting interrupt flag.	0x00: Converted value has zero deviation 0x01: Converted value has up to 1 LSB of error 0x02: Converted value has up to 2 LSBs of error ... 0xFF: Converted value has up to 255 LSBs of error

**REGADCBIST5 (0x1D33)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFLIMSCL2[7:0]							
<b>Reset</b>	0x07							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
REFLIMSCL 2	7:0	Control the accuracy for 1/4 scale measurement. Sets maximum error from expected ADC conversion code in LSBs before asserting interrupt flag.	0x00: Converted value has zero deviation 0x01: Converted value has up to 1 LSB of error 0x02: Converted value has up to 2 LSBs of error ... 0xFF: Converted value has up to 255 LSBs of error

**REGADCBIST6 (0x1D34)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	REFLIMSCL3[7:0]							
<b>Reset</b>	0x07							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
REFLIMSCL 3	7:0	Control the accuracy for 1/8 scale measurement. Sets maximum error from expected ADC conversion code in LSBs before asserting interrupt flag.	0x00: Converted value has zero deviation 0x01: Converted value has up to 1 LSB of error 0x02: Converted value has up to 2 LSBs of error ... 0xFF: Converted value has up to 255 LSBs of error

**REGADCBIST7 (0x1D35)\***

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TLIMIT[7:0]							
<b>Reset</b>	0x03							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
TLIMIT	7:0	Control accuracy for alternate temperature sensor measurement. Sets maximum deviation between primary and alternate temperature sensors before asserting interrupt flag.	0x00: Zero difference allowed between sensors 0x01: 0.5 to 1 degree K difference between sensors 0x02: 0.5 to 1 degree K difference between sensors 0x03: 2 degree K difference between sensors 0x04: 2.5 to 3 degree K difference between sensors 0x05: 3 to 3.5 degree K difference between sensors 0x06: 3.5 to 4 degree K difference between sensors 0x07: 4.5 to 5 degree K difference between sensors Others: Reserved

[REGADCBIST9 \(0x1D37\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MUXV_CTRL[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
MUXV_CTRL	7:0	Value to drive GPIO during MUX testing. This is driven by BIST engine, but is provided as a hook to drive during debug.	0b11111110: Drive ADC0 MUX input to Low (GND) and all others to High (ADC Voltage Reference) 0b00000001: Drive ADC0 MUX input to High (ADC Voltage Reference) and all others to Low (GND) 0b11111101: Drive ADC1 MUX input to Low (GND) and all others to High (ADC Voltage Reference) 0b00000010: Drive ADC1 MUX input to High (ADC Voltage Reference) and all others to Low (GND) 0b11111011: Drive ADC2 MUX input to Low (GND) and all others to High (ADC Voltage Reference) 0b00000100: Drive ADC2 MUX input to High (ADC Voltage Reference) and all others to Low (GND)

[REGADCBIST12 \(0x1D3A\)\\*](#)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TMONCAL_OOD_WAIT_B2[7:0]							
<b>Reset</b>	0xFF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS	DESCRIPTION	DECODE					
TMONCAL_OOD_WAIT_B2	7:0	Count value to set time before temperature sensor Calibration goes out-of-date. 1 LSB is 26.2ms.	0x00: 26.2ms 0x0F: 418.7ms 0xFE: 6.7s 0xFF: disabled					

REGADCBIST13 (0x1D3B)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	T_EST_OUT_B0[7:0]							
<b>Reset</b>	0xFF							
<b>Access Type</b>	Read Only							
BITFIELD	BITS			DESCRIPTION				
T_EST_OUT_B0	7:0			Bits 7:0 of temperature sensor measurement. Reconstruct temperature using this register with T_EST_OUT_B1. Bits 9:0 divided by 2 is the die junction temperature in Kelvin.				

REGADCBIST14 (0x1D3C)

BIT	7	6	5	4	3	2	1	0		
<b>Field</b>	T_EST_OUT_B1[1:0]		–	–	–	–	ALT_T_EST_OUT_B1[1:0]			
<b>Reset</b>	0x3		–	–	–	–	0x3			
<b>Access Type</b>	Read Only		–	–	–	–	Read Only			
BITFIELD	BITS			DESCRIPTION						
T_EST_OUT_B1	7:6			Bits 9:8 of temperature sensor measurement. Reconstruct temperature using this register with T_EST_OUT_B0. Bits 9:0 divided by 2 is the die junction temperature in Kelvin.						
ALT_T_EST_OUT_B1	1:0			Bits 9:8 of temperature sensor measurement. Reconstruct temperature using this register with ALT_T_EST_OUT_B0. Bits 9:0 divided by 2 is the die junction temperature in Kelvin.						

REGADCBIST15 (0x1D3D)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ALT_T_EST_OUT_B0[7:0]							
<b>Reset</b>	0xFF							
<b>Access Type</b>	Read Only							
BITFIELD	BITS			DESCRIPTION				
ALT_T_EST_OUT_B0	7:0			Bits 7:0 of alternate temperature sensor measurement. Reconstruct temperature using this register with ALT_T_EST_OUT_B1. Bits 9:0 divided by 2 is the die junction temperature in Kelvin.				

REGCRC8 (0x1D40)

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SKIP0_LSB[7:0]							
<b>Reset</b>	0xFF							
<b>Access Type</b>	Write, Read							
BITFIELD	BITS			DESCRIPTION				
SKIP0_LSB	7:0			Address 0 to skip (LSB)				

[REGCRC9 \(0x1D41\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP0_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP0_MSB	7:0		Address 0 to skip (MSB)					

[REGCRC10 \(0x1D42\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP1_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP1_LSB	7:0		Address 1 to skip (LSB)					

[REGCRC11 \(0x1D43\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP1_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP1_MSB	7:0		Address 1 to skip (MSB)					

[REGCRC12 \(0x1D44\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP2_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP2_LSB	7:0		Address 2 to skip (LSB)					

[REGCRC13 \(0x1D45\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP2_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP2_MSB	7:0		Address 2 to skip (MSB)					

[REGCRC14 \(0x1D46\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP3_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP3_LSB	7:0		Address 3 to skip (LSB)					

[REGCRC15 \(0x1D47\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP3_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP3_MSB	7:0		Address 3 to skip (MSB)					

[REGCRC16 \(0x1D48\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP4_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP4_LSB	7:0		Address 4 to skip (LSB)					

[REGCRC17 \(0x1D49\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP4_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP4_MSB	7:0		Address 4 to skip (MSB)					

[REGCRC18 \(0x1D4A\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP5_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP5_LSB	7:0		Address 5 to skip (LSB)					

[REGCRC19 \(0x1D4B\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP5_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP5_MSB	7:0		Address 5 to skip (MSB)					

[REGCRC20 \(0x1D4C\)](#)

BIT	7	6	5	4	3	2	1	0
Field	SKIP6_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP6_LSB	7:0		Address 6 to skip (LSB)					

REGCRC21 (0x1D4D)

BIT	7	6	5	4	3	2	1	0
Field	SKIP6_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP6_MSB	7:0		Address 6 to skip (MSB)					

REGCRC22 (0x1D4E)

BIT	7	6	5	4	3	2	1	0
Field	SKIP7_LSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP7_LSB	7:0		Address 7 to skip (LSB)					

REGCRC23 (0x1D4F)

BIT	7	6	5	4	3	2	1	0
Field	SKIP7_MSB[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
SKIP7_MSB	7:0		Address 7 to skip (MSB)					

CC\_RTTN\_ERR (0x1D5F)\*

BIT	7	6	5	4	3	2	1	0		
Field	—	—	—	—	—	RESET_EF USE_CRC_	INJECT_EF USE_CRC_	INJECT_RT TN_CRC_E		
Reset	—	—	—	—	—	0b0	0b0	0b0		
Access Type	—	—	—	—	—	Write Clears All, Read	Write, Read	Write, Read		
BITFIELD	BITS		DESCRIPTION				DECODE			
RESET_EF USE_CRC_	2	Reset eFuse CRC error status to 0. Self-clearing. Use for ASIL evaluation purposes.				0b0: Do not reset 0b1: Reset				
INJECT_EF USE_CRC_	1	Set this bit before reading eFuse values to inject error to eFuse CRC value (EFUSE_CRC_ERR bit). Use for ASIL evaluation purposes.				0b0: Do not inject eFuse CRC errors 0b1: Inject eFuse CRC errors				

BITFIELD	BITS	DESCRIPTION	DECODE
INJECT_RTT_N_CRC_ER	0	Set this bit before going into sleep mode to inject error to retention memory CRC value (RTTN_CRC_INT bit). Use for ASIL evaluation purposes.	0b0: Do not inject retention memory CRC errors 0b1: Inject retention memory CRC errors

## Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	12/20	Release for Market Intro	—
1	12/21	Corrected ADC input voltage range when using /3 and /4 dividers	13
		Documented typical minimum PWDNB hold time	14
		Added guidance on reference clock input rise and fall times	18
		Added to pin descriptions for improved accuracy and consistency of terminology throughout document. Added documentation for video timing generator I/O	21-24
		Added note to recommended operation conditions table to clarify definition of supply noise	26
		Updated typical operating conditions for VDDIO current vs. GPIO toggle rate plot with corrected data	26
		Added $R_{TERM}$ to external components table. Was previously missing	27
		Updated GMSL2 Lock Time figure	29
		Added Additional Documentation section	26
		Provided more detail on tunneling mode for clarity	36-38
		Corrected frequency deviations in Spread Spectrum section. These were inconsistent with the (correct) register map	43
		Added detail in video timing generator section	38
		Clarified that video crossbar is only available in pixel mode	39
		Corrected details in GMSL2 Bandwidth Calculations Section and added figure 23 for clarity	46
		Documented typical GPIO delay for GPIO forwarding from serializer to deserializer in non-compensated mode	38
		Added detail on functional safety features	49
		Corrected errors, added power-up defaults and video timing generator functions to MFP Pin Function Map	58-59
		Thermal management section rewritten for accuracy	64
		Removed reference to TMON in Junction Temperature Monitoring and Overtemperature Alarm section. User should use built-in precision temperature sensor for accurate die temperature measurements	54
		Revised recommendation for power supply ramp time	62
		Removed requirement for clock stretching on remote side	64
		Added software programming model section	64
		Added typical application circuit	73
		Added table to show typical maximum cable lengths and wording to reference GMSL2 Channel Specification	1, 41
		Changed wording, fixed typos, reordered figures and sections for improved accuracy, clarity and logical grouping	1-230
		Register Map updates: Global: Added Reset column, added RSVD for hidden bits, added Note and asterisks to show which registers are retained in sleep mode, and changed register descriptions and decodes for clarity.	

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
		Unhid registers CTRL2 (0x12), CMU (0x302), EXT21 (0x38D), EXT22 (0x38E), EXT23 (0x38F), REF_VTG (0x3F1), RLMS17 (0x14), RLMS1C (0x141C), RLMS1D (0x141D), RLMS1F (0x141F), RLMS32 (0x1432), RLMS3A (0x143A), RLMSA8 (0x14A8), RLMSA9 (0x14A9), DPLL_0 (0x1A00), DPLL_7 (0x1A07), REGADCBIST3-REGADCBIST7 (0x1D32-0x1D35), REGADCBIST9 (0x1D37)	
		Hid registers PWR1 (0x9), TX1 (0x59), TR1 (0x79, 0x81, 0x89, 0x91, 0xA1, 0xA9), TR0 (0x88), EXT9 (0x381), EXT10 (0x383), PM_OV_STAT2 (0x572), PM_OV_STAT3 (0x573), RLMS30 (0x1430), RLMS31 (0x1431), RLMS32 (0x1460), RLMSA6 (0x14A6), RLMSD8 (0x14D8), RLMSE1 (0x14E1), DPLL_11 (0x1A0B), I2C_UART_CRC5 (0x1D0D), I2C_UART_CRC6 (0x1D0E), MEM_ECC1 (0x1D15), MEM_ECC2 (0x1D16), REGADCBIST2 (0x1D30), REGADCBIST2 (0x1D30), REGADCBIST8 (0x1D36), REGADCBIST10 (0x1D38) REGADCBIST11 (0x1D39).	
		Unhid bitfields VREF_CAP_EN (0x11), config_sel_clock_out_use_external (0x1A03), config_spread_bit_ratio (0x1A03), config_div_fb_exp (0x1A09), config_allow_coarse_change (0x1A0A), config_div_out_exp (0x1A0A). MUXVER_EN (0x1D28).	
		Hid bitfields PHY_INT_OEN_A (0x1A), PHY_INT_A (0x1B), VPRBS_ERR_OEN (0x1E), EFUSE_CRC_ER (0x1F), RVRBS_ERR_FLAG (0x1F), TIMEOUT (0x2B), GPIO_TX_WNDW (0x31)ARB_TO_LEN (0x48), PRIO_VAL (0x58, 0x78, 0x88, 0x90, 0xA0, 0xA8), PRIO_CFG (0x58, 0x78, 0x88, 0x90, 0xA0, 0xA8), VID_PRBS_CHK_EN (0x26B), DIS_COLOR_CROSSBAR (0x276), CLK_SELZ (0x308), phy_config (0x330), t_hs_dec_en (0x338), cphy_hdr_err; cphy_hdr1_err; cphy_hdr2_err; InvCode_In1; InvCode_In0 (0x380), CPhy_Mode; phy1_CPhyCdrTp_bt; phy1_CPhyCdrSwap_en; phy1_CPhyCdrMask (0x383), adc_refs cl (0x501), bypass_volttemp_corr (0x509), adc_over flow_ie (0x50C), gpio_err_ie (0x50F), adc_over flow_if (0x510), gpio_err_if (0x513), invalid_ch_sel_if (0x513), PIO09_SLEW (0x571), EOM_RST_THR (0x1406), MAN_CALIB (0x14A7), MSGCNTR_ERR_THR (0x1D09), CRC_ERR_THR (0x1D09), MEM_ECC_ERR2_THR (0x1D14), MEM_ECC_ERR1_THR (0x1D14), POST_RUN_MBIST (0x1D20), POST_RUN_LBIST (0x1D20), RR_MUX (0x1D28), RUN_MUX (0x1D28).	
2	10/22	Added Min HOLD time	14
		Corrected ADC conditions column when using /3 and 34 dividers	13
		Added Maximum Rise and Fall times in EXTERNAL CLOCK INPUT ON X1, X2 FLOATING section	18
		Added VTG pins to the MFP Pin Function Map	58
		Moved the Operating Junction Temperature from TYP to MAX column	26
		Added note for the Maximum Supply Noise in the Recommended Operating Conditions section	26
		Added description of crystal load capacitors	27
		Added description of VDDIO decoupling capacitors	27
		Updated GMSL2 Lock Time Figure	29
		Updated I2C Timing Figure	30
		Updated format of figures and content	1-234
3	1/23	Updated Tunneling and Pixel Mode section.	37-38
4	5/23	Updated GMSL Bandwidth Calculations	46
		Updated I2C/UART verbiage	47, 66
		Updated TYP Lock time to 45ms and updated Link Lock Section	15, 44-45
		Updated Device Reset Section	59
		Updated SION pin description	21

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
		Removed mandatory LDO write from programming notes section	76
		Added comment about adding 1MΩ pull-down on ERRB pin	61
		Removed comment about supporting GPIO transitions of 1MHz	48, 54
		Added comment about ADI supplying PMICs that support MAX96717	61
		Updated SPI section	48
		Updated Note 2 below EC Table	19
		Moved ESD table out of EC Table	19
		Updated thermal guidance verbiage	10
		Updated ADC EC Table	13-14
		Added a note about using internal bandgap reference for high accuracy ADC measurements	50
		Register 0x0E, DEV_REV changed from 0x04 to 0x06	97
5	9/23	Register Map: Updated Device ID	96
5	9/23	Register Map: Updated Reset/Default value for Register 0x1417 from F9 to 0.	211
5	9/23	Register Map: Updated Reset/Default value for Register 0x1432 from FF to 7F.	212
5	9/23	Programming Notes: Removed mandatory register write for 0x302.	66
6	3/25	Moved sections for better understanding. Corrected grammatical, formats/styles, and spelling.	Many
6	3/25	Top Marking = MAXIM column removed from Ordering Information table.	75
6	3/25	Corrected typo in CAP_VDD absolute maximum rating (0.3 to 1.2V changed to -0.3 to 1.2 V).	10
6	3/25	Updated figure numbers and formatting for Typical Operating Characteristics curves.	21
6	3/25	Removed Video Timing Generator (VTG) from text and MFP table.	23-24, 39, 60
6	3/25	SSC PPM value corrected from ±1250 to ±1265 PPM.	61
6	3/25	Removed asterisks for registers 0x330-0x335, and 0x345-346, clarifying that they are not stored in retention memory.	84-85
6	3/25	Added statement in minimum blanking section suggesting use of the GMSL GUI bandwidth calculator.	57
6	3/25	Updated Coax PoC layout example figure with differential 100Ω length-matched traces for SION and SIOP.	64
6	3/25	Corrected description of number of MFP pins available.	49, 55
6	3/25	Changed wording in Minimum Blanking Section ("blanking period" to "sync width").	57
6	3/25	Removed TX_Rate options that do not apply in register 0x01 description.	95
6	3/25	Corrected Access Type description for some bits in registers 0x10, 0x1D, 0x510-0x513, 0x1D28.	100, 104, 193-195, 237
6	3/25	Corrected details of default settings for MFP1 and MFP2, MFP7, and MFP8.	23-24, 59
6	3/25	Added description of Unique Serial Number feature.	65
6	3/25	Added statement about register 0x302 in Programming Notes section.	66
6	3/25	Register 0x344 decode updated for unsupported data type case.	172
6	3/25	Package code truncated.	10
6	3/25	Added decode to Register 0x1D35.	241
6	3/25	Added statement to not read the ADC conversion done flag before the conversion is done.	238

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
		ADC Input Select for ADC Output Register decode updated for 0xB: TMON.	196, 198-200, 202-204, 206
		Updated ADC description to state that the default 1.25 V voltage reference is preferred.	51-52
		Updated ADC description to state that the input buffer should always be used and not bypassed.	52
		Updated description of bit 5 in register 0x112 to clarify that it is only applicable in pixel mode.	123
		Updated description of bit 0 in register 0x380 to clarify that it is only applicable in tunnel mode.	177
		Updated Typical Application Circuit to show pull-up resistors for I2C/UART.	74
		Unhid some or all bits in registers 0x02, 0x03, 0x07, 0x09, 0x88, 0x8B, 0x8E, 0x8F, 0x0F, 0x337, 0x536, 0x537, 0x509, 0x2C0, 0x2C3, 0x2C6, 0x2C9, 0x2CC, 0x2CF, 0x2D2, 0x2D5, 0x2D8, 0x2DB, 0x2DE, 0xD00-0xD03, 0xD40-0xD4F, 0x1D15-0x1D18, 0x1403, 0x1484, 0x1485, 0x1495, 0x14A8, 0x14A9, 0x14BA, 0x14C8, 0x14C9, 0x14CA, and 0x1A03.	Many
		Hid registers 0x3F1-0x3F3.	187
7	7/25	Package Information: Updated Package Code column with note to click on Land Pattern Number link.	10

#### MIPI NOTICE OF DISCLAIMER

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

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

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

Without limiting the generality of this Disclaimer stated above, the user of the contents of this Document is further notified that MIPI: (a) does not evaluate, test or verify the accuracy, soundness or credibility of the contents of this Document; (b) does not monitor or enforce compliance with the contents of this Document; and (c) does not certify, test, or in any manner investigate products or services or any claims of compliance with the contents of this Document. The use or implementation of the contents of this Document may involve or require the use of intellectual property rights ("IPR") including (but not limited to) patents, patent applications, or copyrights owned by one or more parties, whether or not Members of MIPI. MIPI does not make any search or investigation for IPR, nor does MIPI require or request the disclosure of any IPR or claims of IPR as respects the contents of this Document or otherwise. Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to: MIPI Alliance, Inc. c/o IEEE-ISTO 445 Hoes Lane Piscataway, NJ 08854 Attn: Board Secretary