DisplayPort Driver API Reference

Technical Note

FPGA-TN-02423-1.1

June 2026

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Contents 3
Abbreviations in This Document 5
1. Introduction 6
1.1. Purpose 6
1.2. Audience 6
1.3. Driver Versioning 6
1.3.1. Driver Version 6
1.4. Driver and IP Compatibility 6
2. API Description 7
2.1. dp_sink_init() 7
2.2. dp_sink_cmd() 7
2.3. dp_sink_get_msa_lock() 7
2.4. dp_sink_get_msa_updated() 7
2.5. dp_sink_read_config() 8
2.6. dp_src_init() 8
2.7. dp_src_msa_set() 8
2.8. dp_src_read_config() 8
3. Function Call Flow Diagrams 9
3.1. dp_sink_init() 9
3.2. dp_sink_cmd() 10
3.3. dp_sink_get_msa_lock() 11
3.4. dp_sink_get_msa_updated() 12
3.5. dp_sink_read_config() 13
3.6. dp_src_init() 14
3.7. dp_src_msa_set() 15
3.8. dp_src_read_config() 16
4. API Data Structures 17
4.1. struct src_reg_map_t 17
4.2. struct sink_reg_map_t 17
4.3. dp_video_timing_t 17
4.4. dp_config_t 18
4.5. dp_status_t 18
4.6. dp_src_handle_t 18
4.7. dp_sink_handle_t 19
5. API Enum 20
5.1. enum dp_src_cmd_st 20
5.2. enum dp_link_rate_t 20
5.3. enum dp_lane_count_t 20
5.4. enum dp_color_depth_t 20
5.5. enum dp_color_format_t 21
5.6. enum dp_sink_cmd_st 21
6. API Macros 22
References 27
Technical Support Assistance 28
Revision History 29

Figures

Figure 3.1. unsigned char dp_sink_init() 9
Figure 3.2. unsigned char dp_sink_cmd() 10
Figure 3.3. unsigned char dp_sink_get_msa_lock() 11
Figure 3.4. unsigned char dp_sink_get_msa_updated() 12
Figure 3.5. dp_config_t* dp_sink_read_config() 13
Figure 3.6. unsigned char dp_src_init() 14
Figure 3.7. unsigned char dp_src_msa_set() 15
Figure 3.8. dp_config_t* dp_src_read_config() 16

Tables

Table 1.1. Driver and IP Compatibility 6
Table 4.1. src_reg_map_t Parameters 17
Table 4.2. sink_reg_map_t Parameters 17
Table 4.3. dp_video_timing_t Parameters 17
Table 4.4. dp_config_t Parameters 18
Table 4.5. dp_status_t Parameters 18
Table 4.6. dp_src_handle_t Parameters 18
Table 4.7. dp_sink_handle_t Parameters 19
Table 5.1. dp_src_cmd_st Variables 20
Table 5.2. dp_link_rate_t Variables 20
Table 5.3. dp_lane_count_t Variables 20
Table 5.4. dp_color_depth_t Variables 20
Table 5.5. dp_color_format_t Variables 21
Table 5.6. dp_sink_cmd_st Variables 21
Table 6.1. API Macros 22

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviations	Definition
API	Application Programming Interface
BER	Bit Error Rate
DP	DisplayPort
DPCD	DisplayPort Configuration Data
DP_SINK	DisplayPort Receiver
DP_SRC	DisplayPort Transmitter
DVI	Digital Visual Interface
FPGA	Field Programmable Gate Array
HDTV	High-Definition Television
HPD	Hot Plug Detect
IP	Intellectual Property
IRQ	Interrupt Request
LT	Link Training
MSA	Main Stream Attributes
MSB	Most Significant Bit
PC	Personal Computer
PHY	Physical Layer
QSPI	Quad Serial Peripheral Interface
RX	Receiver
TU	Transfer Unit
TX	Transmitter
VBID	Video Blanking Interval Data
VESA	Video Electronics Standards Association
VGA	Video Graphics Array
VOD	Voltage Output Differential

1. Introduction

The Lattice™ DisplayPort™ IP core is designed for transmission and reception of serial-digital video for consumer and professional displays. This IP helps you to implement a DisplayPort video interface as defined by the Video Electronics Standards Association (VESA) DisplayPort specifications. DisplayPort is a high-speed serial interface standard supported by industry leaders in consumer electronics high-definition television (HDTV), personal computer (PC) laptop, and PC monitors. This protocol is considered a successor to Video Graphics Array (VGA) and Digital Visual Interface (DVI) standards with support for video resolutions up to 4K video and multi-channel audio.

Refer to the DisplayPort IP User Guide (FPGA-IPUG-02236) for more details about the IP core.

1.1. Purpose

This document is intended to act as a reference guide for developers by providing details of the C language driver APIs and function call flows.

1.2. Audience

The intended audience for this document includes embedded system designers and embedded software developers using CertusPro™-NX, Lattice Avant™ (Avant-G, Avant-X), Certus™-N2 (except LN2-CT-20ES) devices. The technical guide assumes readers have expertise in embedded systems and FPGA technologies.

1.3. Driver Versioning

1.3.1. Driver Version

DISPLAYPORT_DRV_VER “26.1.0”

1.4. Driver and IP Compatibility

Table 1.1. Driver and IP Compatibility

Driver Version	IP Version
26.1.0	2.2.0

Refer to the DisplayPort IP Release Notes (FPGA-RN-02071) for more information on the driver and IP versions.

2. API Description

2.1. dp_sink_init()

This API sets the base address for the DisplayPort receiver in the provided handle.

```
unsigned char dp_sink_init(dp_handle_t *handle, unsigned int base_address)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_handle_t structure.	0: success 1: failure
In	base_address	Base address to be assigned to DP receiver.	

2.2. dp_sink_cmd()

This API dispatches various DP sink commands based on the specified command type.

```
unsigned char dp_sink_cmd(dp_handle_t *handle, dp_sink_cmd_st cmd, unsigned int *val)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_handle_t structure.	0: success 1: failure
In	cmd	DP sink command for execution (of type dp_sink_cmd_st).	
In/Out	val	Pointer to an unsigned int used as input or output depending on the command.	

2.3. dp_sink_get_msa_lock()

This API reads the main stream attribute (MSA) lock bit from the DP_SINK_VBID_MSA_STS_REG register.

The lock bit indicates whether the MSA data is locked or stable.

```
unsigned char dp_sink_get_msa_lock(const dp_sink_handle_t *handle, unsigned int *val)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_sink_handle_t structure.	0: success 1: failure
Out	val	Pointer to an unsigned int where the MSA lock status is stored. Must not be NULL. Returns 1 if MSA is locked, 0 otherwise.	

2.4. dp_sink_get_msa_updated()

This API reads the MSA updated bit from the DP_SINK_VBID_MSA_STS_REG register.

The updated bit indicates whether the MSA data is updated.

```
unsigned char dp_sink_get_msa_updated(const dp_sink_handle_t *handle, unsigned int *val)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_sink_handle_t structure.	0: success 1: failure
Out	val	Pointer to an unsigned int where the MSA updated status is stored. Must not be NULL. Returns 1 if MSA has been updated, 0 otherwise.	

2.5. dp_sink_read_config()

This API reads the DisplayPort sink configuration from hardware and returns as dp_config_t pointer.

```
const dp_config_t* dp_sink_read_config(dp_sink_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_sink_handle_t structure.	NULL: failure Pointer to the updated configuration on success

2.6. dp_src_init()

This API sets the base address for the DisplayPort transmitter in the provided handle.

```
unsigned char dp_src_init(dp_src_handle_t *handle, unsigned int base_address)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_src_handle_t structure.	0: success 1: failure
In	base_address	Base address to be assigned to DP transmitter.	

2.7. dp_src_msa_set()

This API sets the MSA for the DisplayPort TX using dp_config_t as input.

```
unsigned char dp_src_msa_set(dp_src_handle_t * src_handle, const dp_config_t *config,
unsigned int idx)
```

In/Out	Parameter	Description	Returns
In	src_handle	Handle of the dp_sink_handle_t structure.	0: success 1: failure
In	config	Pointer to dp_config_t configuration. Must not be NULL.	
In	idx	Index value. Reserved for future use, currently unused.	

2.8. dp_src_read_config()

This API reads the DisplayPort src configuration from hardware and returns as dp_config_t pointer.

```
const dp_config_t* dp_src_read_config(dp_src_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	handle	Handle of the dp_src_handle_t structure.	NULL: failure Pointer to the updated configuration on success

3. Function Call Flow Diagrams

3.1. dp_sink_init()

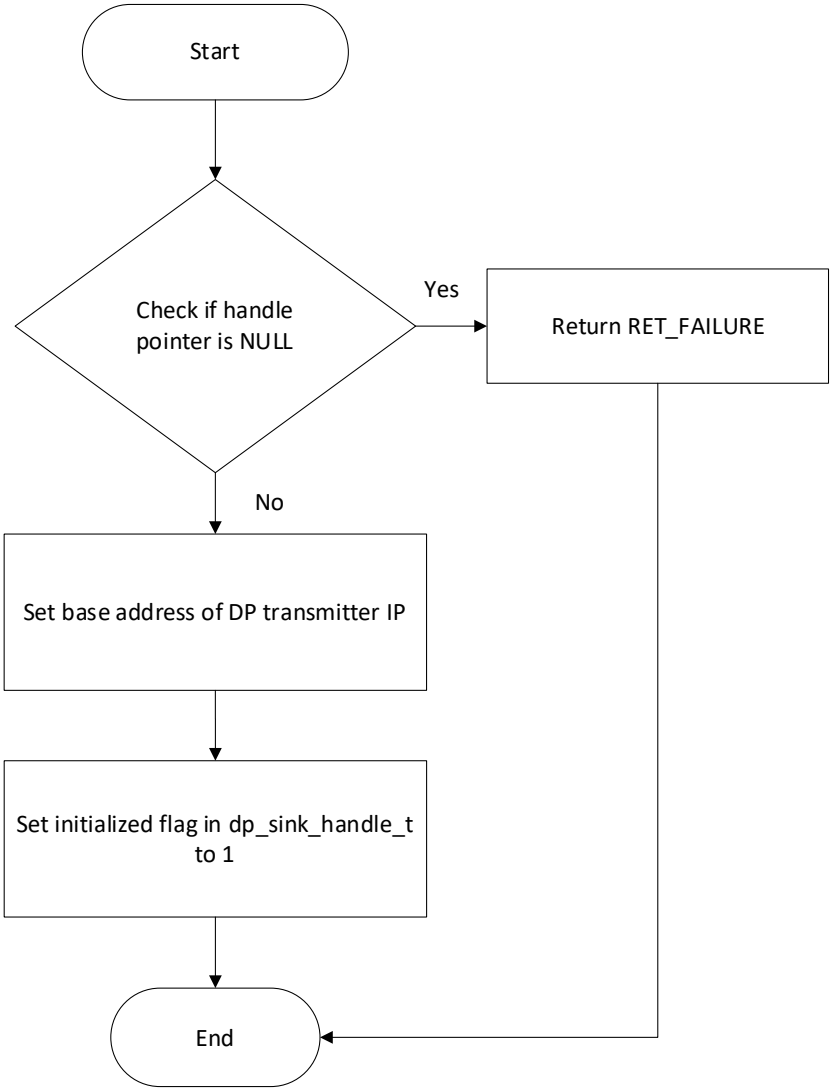

Figure 3.1. unsigned char dp_sink_init()

3.2. dp_sink_cmd()

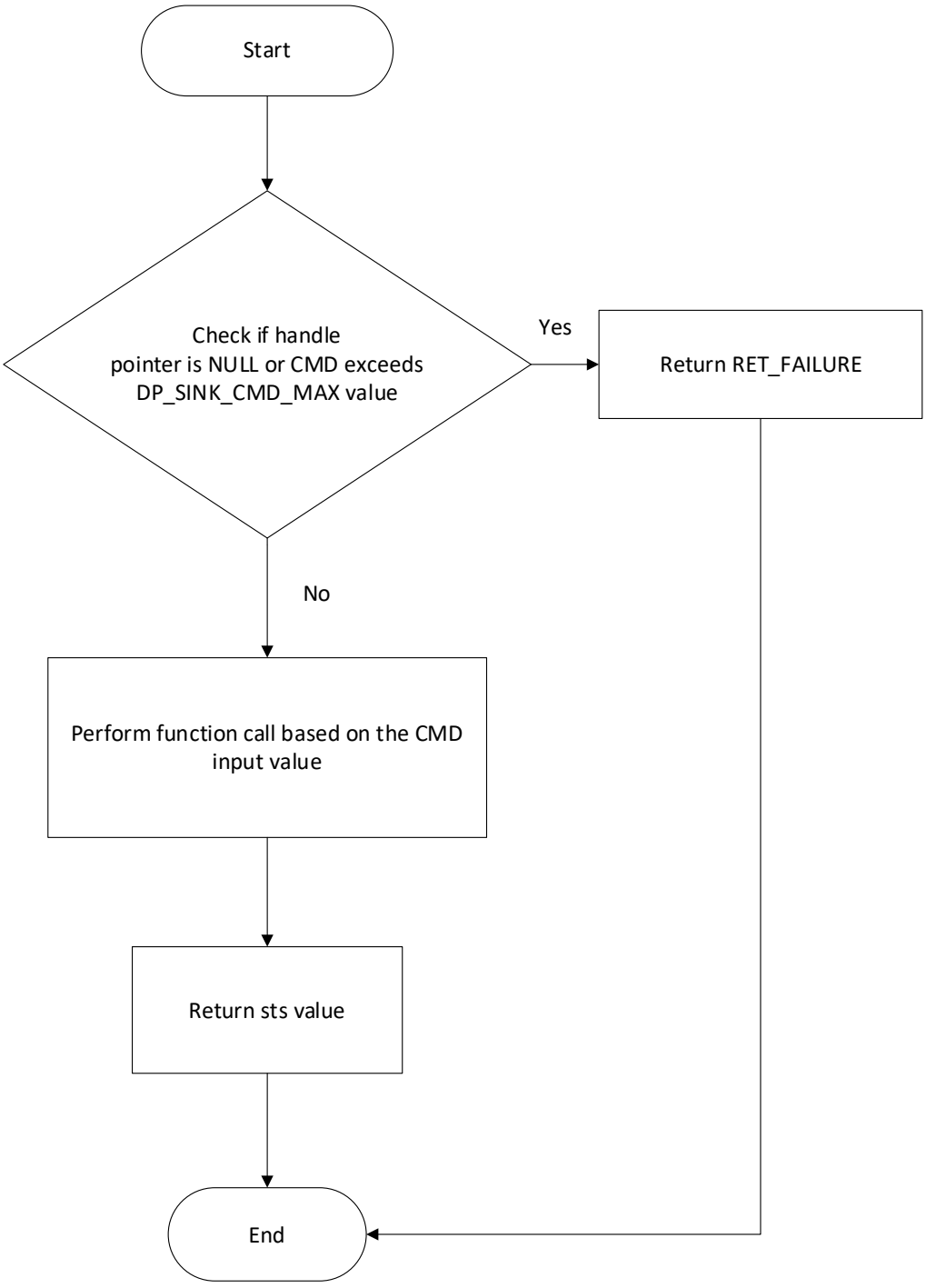

Figure 3.2. unsigned char dp_sink_cmd()

3.3. dp_sink_get_msa_lock()

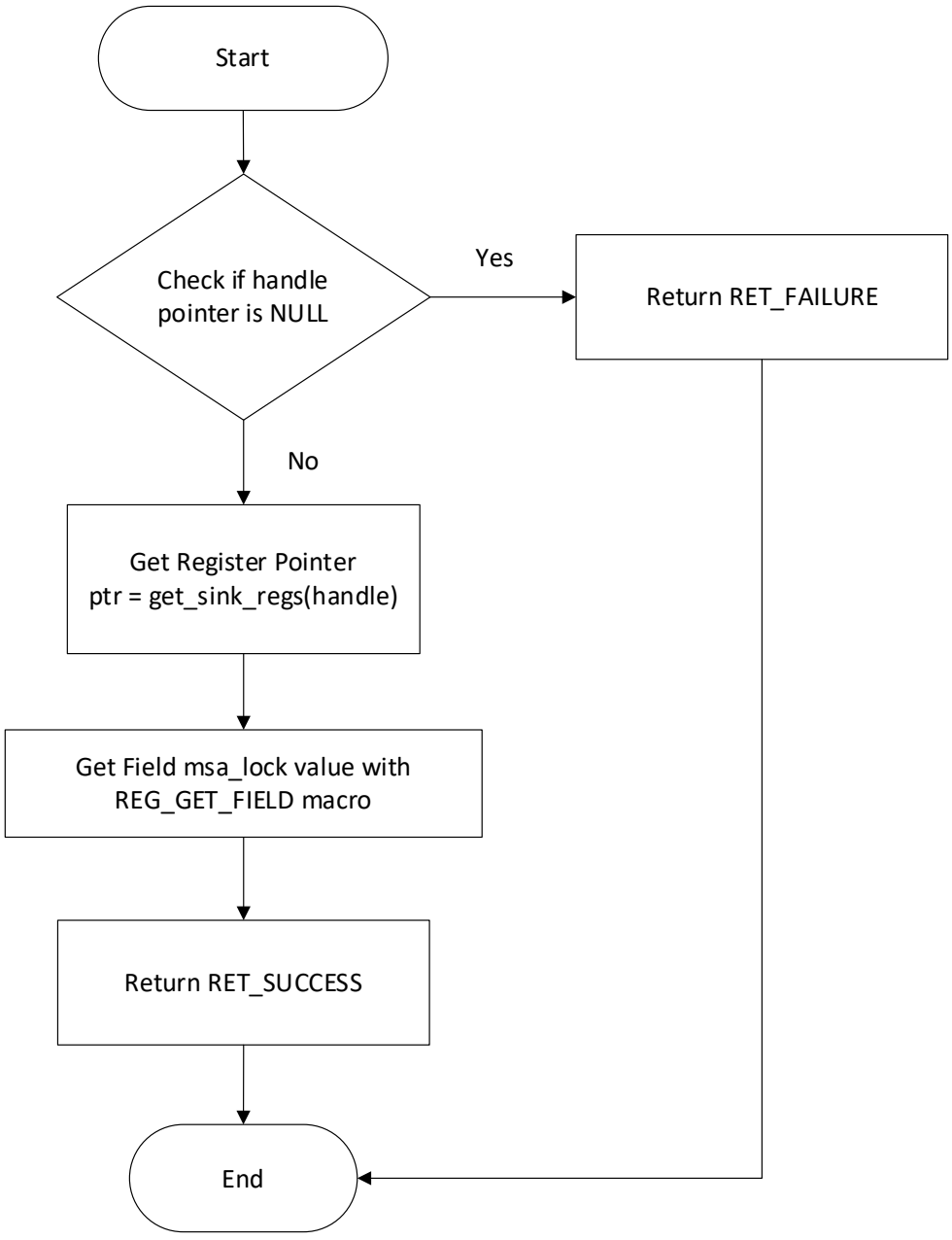

Figure 3.3. unsigned char dp_sink_get_msa_lock()

3.4. dp_sink_get_msa_updated()

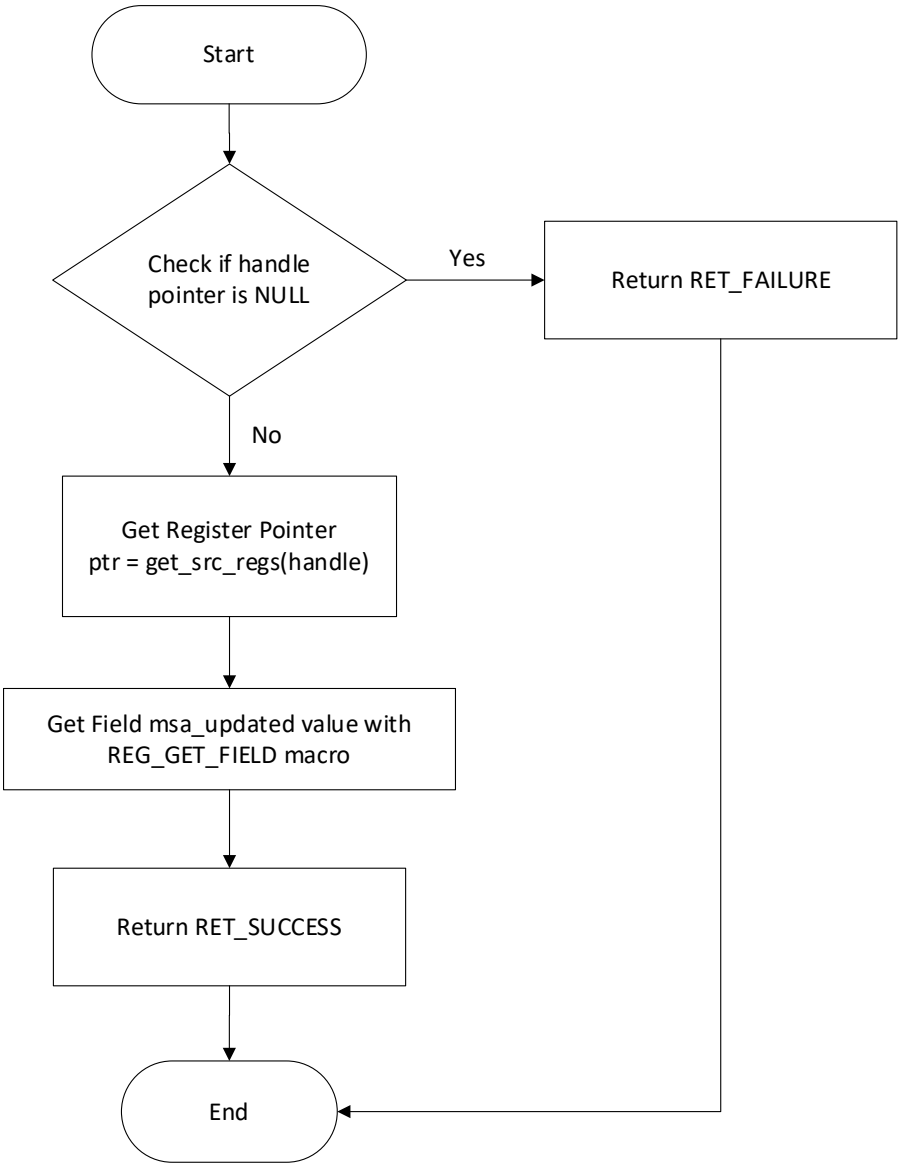

Figure 3.4. unsigned char dp_sink_get_msa_updated()

3.5. dp_sink_read_config()

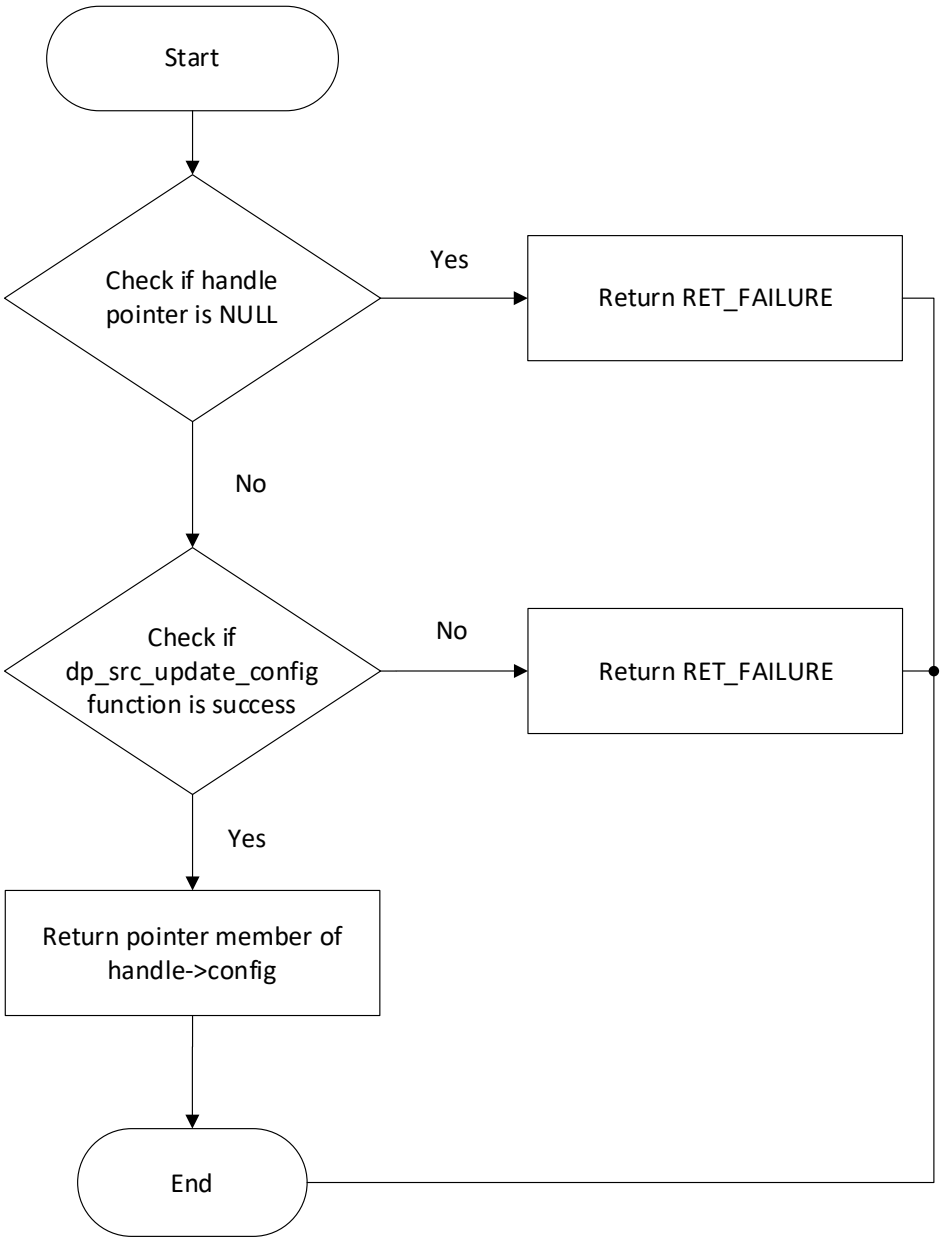

Figure 3.5. dp_config_t* dp_sink_read_config()

3.6. dp_src_init()

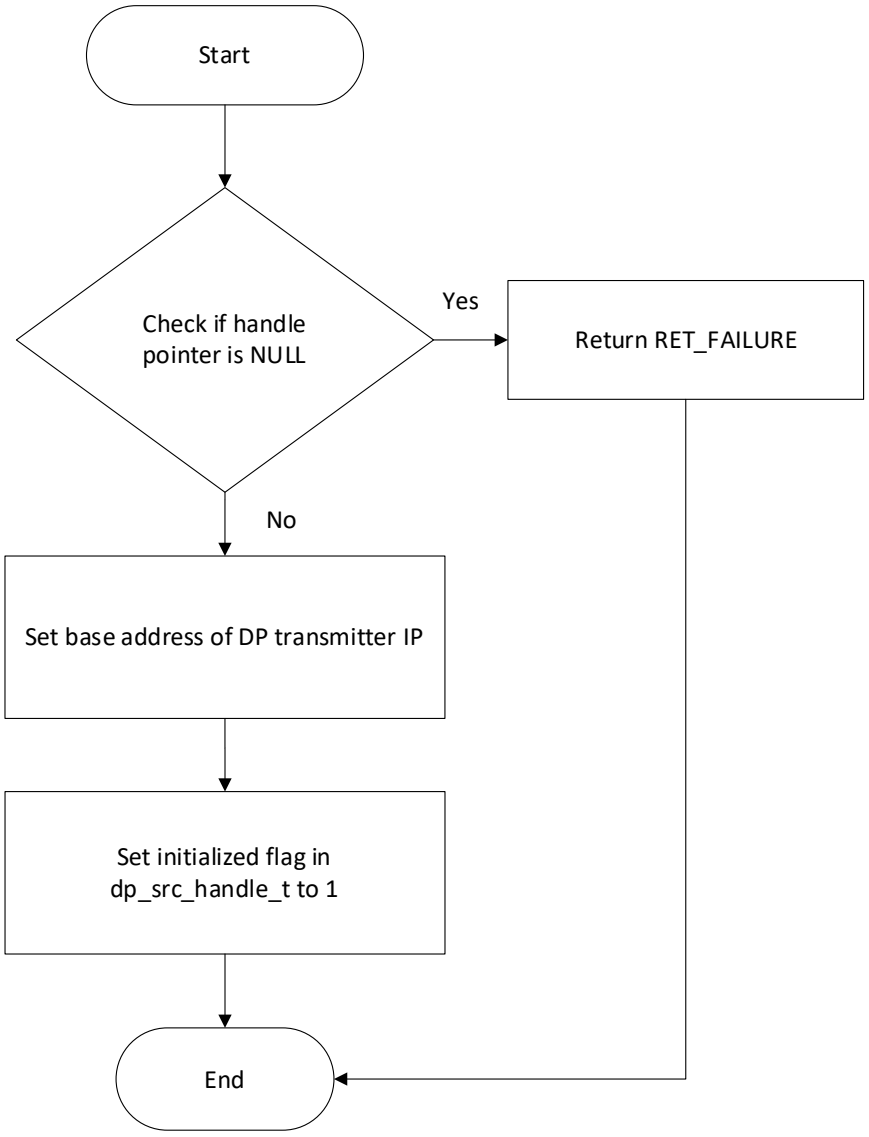

Figure 3.6. unsigned char dp_src_init()

3.7. dp_src_msa_set()

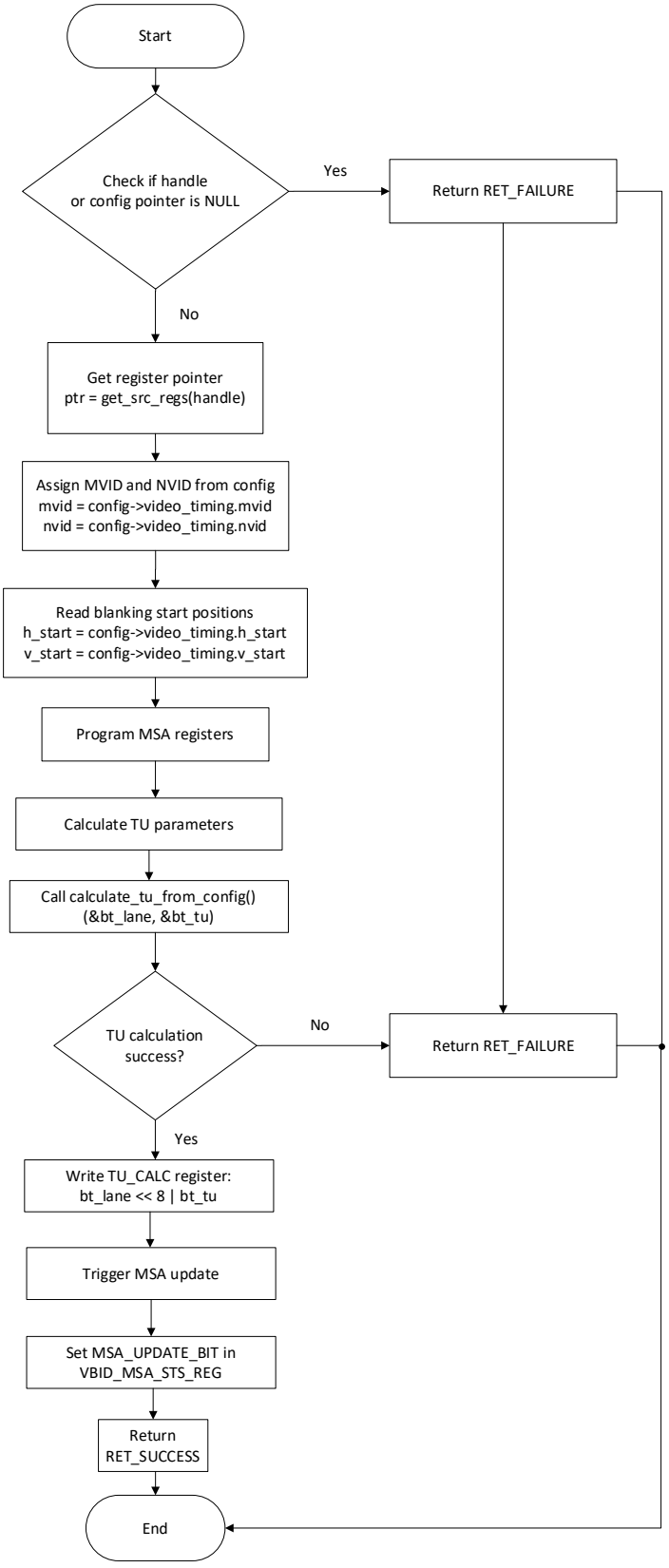

Figure 3.7. unsigned char dp_src_msa_set()

3.8. dp_src_read_config()

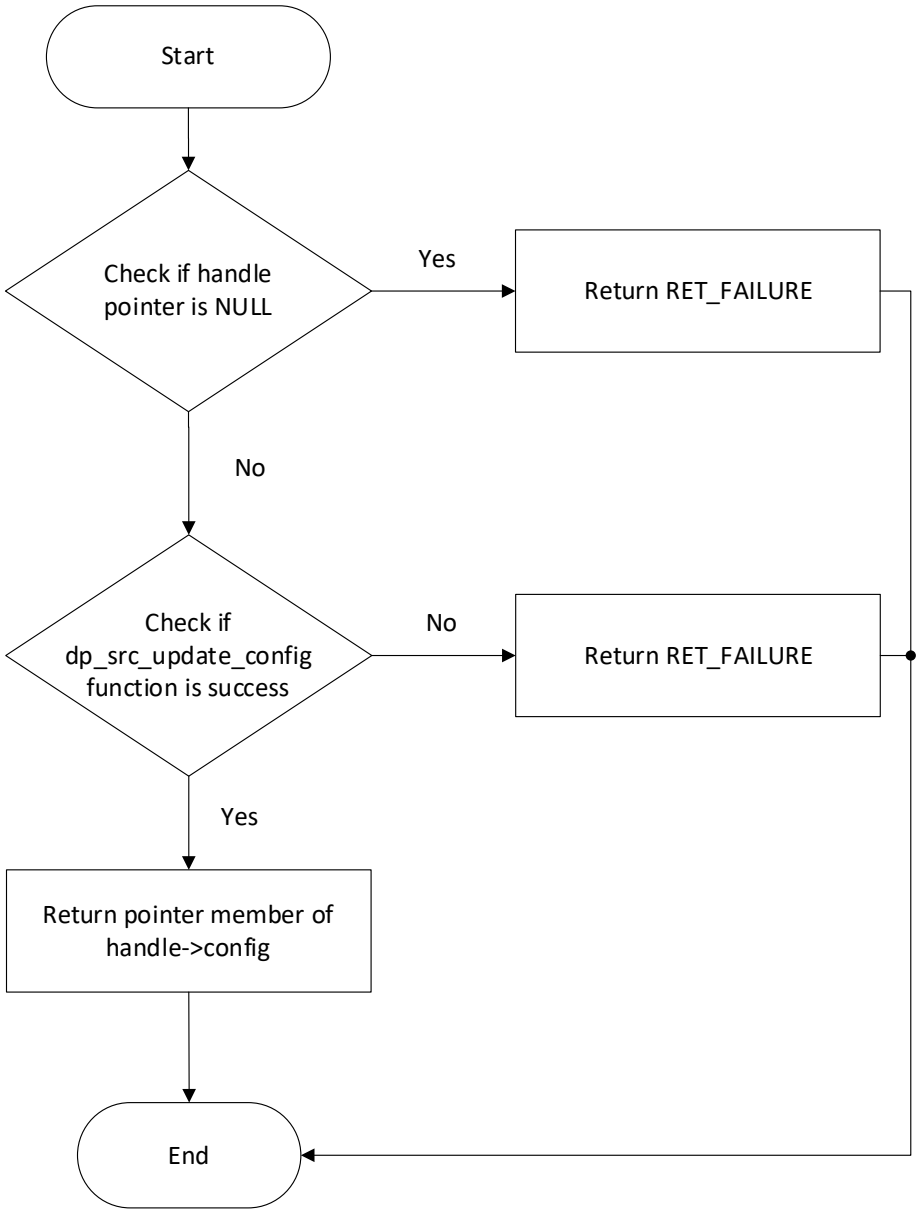

Figure 3.8. dp_config_t* dp_src_read_config()

4. API Data Structures

4.1. struct src_reg_map_t

Table 4.1. src_reg_map_t Parameters

Data Type	Struct Member	Description
volatile unsigned int	DP_SRC_CTRL_REG	DP source control register
volatile unsigned int	DP_SRC_ANALOG_REG	DP source analog register
volatile unsigned int	DP_SRC_PHY_RECONF_REG	DP source PHY reconfiguration register
volatile unsigned int	DP_SRC_HPD_REG	DP source HPD event register
volatile unsigned int	DP_SRC_MSA0_REG	DP source MSA Field0 register
volatile unsigned int	DP_SRC_MSA1_REG	DP source MSA Field1 register
volatile unsigned int	DP_SRC_MSA2_REG	DP source MSA Field2 register
volatile unsigned int	DP_SRC_MSA3_REG	DP source MSA Field3 register
volatile unsigned int	DP_SRC_MSA4_REG	DP source MSA Field4 register
volatile unsigned int	DP_SRC_MSA5_REG	DP source MSA Field5 register
volatile unsigned int	DP_SRC_MSA6_REG	DP source MSA Field6 register
volatile unsigned int	DP_SRC_VBID_MSA_STS_REG	DP source VBID and MSA status register
volatile unsigned int	DP_SRC_TU_CALC_REG	DP source TU calculation register

4.2. struct sink_reg_map_t

Table 4.2. sink_reg_map_t Parameters

Data Type	Struct Member	Description
volatile unsigned int	DP_SINK_CTRL_REG	DP sink control register
volatile unsigned int	DP_LT_SEQ_STS_REG	DP sink link training sequence status register
volatile unsigned int	DP_LINK_STS_REG	DP sink link status register
volatile unsigned int	DP_HPD_EVENT_REG	DP sink HPD event register
volatile unsigned int	DP_SINK_MSA0_REG	DP sink MSA Field0 register
volatile unsigned int	DP_SINK_MSA1_REG	DP sink MSA Field1 register
volatile unsigned int	DP_SINK_MSA2_REG	DP sink MSA Field2 register
volatile unsigned int	DP_SINK_MSA3_REG	DP sink MSA Field3 register
volatile unsigned int	DP_SINK_MSA4_REG	DP sink MSA Field4 register
volatile unsigned int	DP_SINK_MSA5_REG	DP sink MSA Field5 register
volatile unsigned int	DP_SINK_MSA6_REG	DP sink MSA Field6 register
volatile unsigned int	DP_SINK_VBID_MSA_STS_REG	DP sink VBID and MSA status register

4.3. dp_video_timing_t

Table 4.3. dp_video_timing_t Parameters

Data Type	Struct Member	Description
unsigned int	h_total	Total horizontal pixels
unsigned int	h_active	Active horizontal pixels
unsigned int	h_sync_width	Horizontal sync width
unsigned int	h_sync_pol	Horizontal sync polarity
unsigned int	h_start	Horizontal start pixel

Data Type	Struct Member	Description
unsigned int	v_total	Total vertical lines
unsigned int	v_active	Active vertical lines
unsigned int	v_sync_width	Vertical sync width
unsigned int	v_sync_pol	Vertical sync polarity
unsigned int	v_start	Vertical start pixel
unsigned int	pixel_clock	Pixel clock in Hz
unsigned int	mvid	M value for pixel clock regeneration (24-bit)
unsigned int	nvid	N value for pixel clock regeneration (24-bit)

4.4. dp_config_t

Table 4.4. dp_config_t Parameters

Data Type	Struct Member	Description
dp_link_rate_t (refer to Table 5.2)	link_rate	Link rate (1.62, 2.7, 5.4, 8.1 Gbps)
dp_lane_count_t (refer to Table 5.3)	lane_count	Number of physical lanes (1, 2, or 4)
dp_color_depth_t (refer to Table 5.4)	color_depth	Bits per color component (6, 8, 10, 12, 16)
dp_color_format_t (refer to Table 5.5)	color_format	Color encoding format (RGB, YCbCr422, YCbCr444)
dp_video_timing_t (refer to Table 4.3)	video_timing	Video timing parameters (resolution, refresh rate, and other settings)
unsigned int	enhanced_framing	Enhanced framing mode (0=disabled, 1=enabled)
unsigned int	scrambler_disable	Scrambler control (0=enabled, 1=disabled)
unsigned int	msa6_misc	MSA and MISC values

4.5. dp_status_t

Table 4.5. dp_status_t Parameters

Data Type	Struct Member	Description
unsigned int	link_trained	Link training status (0=not trained, 1=trained successfully)
unsigned int	video_enabled	Video transmission status (0=disabled, 1=enabled)
unsigned int	hpd_detected	Hot Plug Detect status (0=not connected, 1=connected)
unsigned int	current_link_rate	Current operating link rate (refer to Table 5.2)
unsigned int	current_lane_count	Current number of lanes in use
unsigned int	pixel_clock_actual	Actual pixel clock frequency in Hz

4.6. dp_src_handle_t

Table 4.6. dp_src_handle_t Parameters

Data Type	Struct Member	Description
volatile src_reg_map_t* (refer to Table 4.1)	src_base_addr	Base address of SRC register block in memory map

Data Type	Struct Member	Description
dp_config_t (refer to Table 4.4)	config	Current configuration settings
dp_status_t (refer to Table 4.5)	status	Current operational status
unsigned int	initialized	Initialization flag (0=not initialized, 1=initialized)

4.7. dp_sink_handle_t

Table 4.7. dp_sink_handle_t Parameters

Data Type	Struct Member	Description
volatile sink_reg_map_t* (refer to Table 4.2)	sink_base_addr	Base address of sink register block in memory map
dp_config_t (refer to Table 4.4)	config	Current configuration settings
dp_status_t (refer to Table 4.5)	status	Current operational status
volatile unsigned int	initialized	Initialization flag (0=not initialized, 1=initialized)

5. API Enum

5.1. enum dp_src_cmd_st

Table 5.1. dp_src_cmd_st Variables

Enum Member	Enum Decimal Value
DPSRC_SET_LANE_CNT	0
DPSRC_SET_LANE_RATE	1
DPSRC_GET_LANE_CNT	2
DPSRC_GET_LANE_RATE	3
DPSRC_CORE_RESET	4
DPSRC_GET_HPD_STS	5
DPSRC_GET_HPD_EVENT	6
DPSRC_CMD_MAX	7 (sentinel — do not use as a command)

5.2. enum dp_link_rate_t

Table 5.2. dp_link_rate_t Variables

Enum Member	Enum Decimal Value
DP_LINK_RATE_1_62_GBPS	6
DP_LINK_RATE_2_7_GBPS	10
DP_LINK_RATE_5_4_GBPS	20
DP_LINK_RATE_8_1_GBPS	30

5.3. enum dp_lane_count_t

Table 5.3. dp_lane_count_t Variables

Enum Member	Enum Decimal Value
DP_LANE_COUNT_1	1
DP_LANE_COUNT_2	2
DP_LANE_COUNT_4	4

5.4. enum dp_color_depth_t

Table 5.4. dp_color_depth_t Variables

Enum Member	Enum Decimal Value
DP_BPC_6	6
DP_BPC_8	8
DP_BPC_10	10
DP_BPC_12	12
DP_BPC_16	16

5.5. enum dp_color_format_t

Table 5.5. dp_color_format_t Variables

Enum Member	Enum Decimal Value
DP_COLOR_FORMAT_RGB	0
DP_COLOR_FORMAT_YUV422	1
DP_COLOR_FORMAT_YUV444	2

5.6. enum dp_sink_cmd_st

Table 5.6. dp_sink_cmd_st Variables

Enum Member	Enum Decimal Value
DP_SINK_SET_LINK_RATE_CAP	0
DP_SINK_SET_LINK_CNT_CAP	1
DP_SINK_CORE_RESET	2
DP_SINK_HPD_OVERRIDE	3
DP_SINK_HPD_FORCE	4
DP_SINK_HPD_IRQ_EN	5
DP_SINK_CMD_MAX	6 (sentinel — do not use as a command)

6. API Macros

Table 6.1. API Macros

Macro	Description
#define BANDWIDTH_RATIO	Bandwidth ratio value (0.27)
#define TU_SIZE	Transfer unit (TU) size value
#define RET_FAILURE	Function failure return value (1)
#define RET_SUCCESS	Function success return value (0)
#define DPSRC_CTRL_LANE_RATE_SHIFT	Bit position (0) for lane rate field in DP_SRC_CTRL register
#define DPSRC_CTRL_LANE_RATE_MASK	Bit mask (0xFF) for 8-bit lane rate field extraction and insertion
#define DPSRC_CTRL_LANE_CNT_SHIFT	Bit position (8) for lane count field in DP_SRC_CTRL register
#define DPSRC_CTRL_LANE_CNT_MASK	Bit mask (0x7) for 3-bit lane count field extraction and insertion
#define DPSRC_CTRL_ENHANCED_FRAME_SHIFT	Bit position (11) for enhanced framing enable in DP_SRC_CTRL register
#define DPSRC_CTRL_ENHANCED_FRAME_MASK	Bit mask (0x1) for single-bit enhanced framing enable flag
#define DPSRC_CTRL_SCRAMBLER_DIS_SHIFT	Bit position (12) for scrambler disable control in DP_SRC_CTRL register
#define DPSRC_CTRL_SCRAMBLER_DIS_MASK	Bit mask (0x1) for single-bit scrambler disable flag
#define DPSRC_CTRL_TRAINING_PATTERN_SHIFT	Bit position (16) for training pattern selection in DP_SRC_CTRL register
#define DPSRC_CTRL_TRAINING_PATTERN_MASK	Bit mask (0xF) for 4-bit training pattern selection field
#define DPSRC_CTRL_CORE_RESET_SHIFT	Bit position (24) for core reset control in DP_SRC_CTRL register
#define DPSRC_CTRL_CORE_RESET_MASK	Bit mask (0x1) for single-bit core reset control flag
#define DPSRC_ANALOG_VOD_L0_SHIFT	Bit position (0) for Lane 0 Voltage Output Differential (VOD) in DP_SRC_ANALOG register
#define DPSRC_ANALOG_VOD_L0_MASK	Bit mask (0x3) for 2-bit Lane 0 VOD field
#define DPSRC_ANALOG_PREE_L0_SHIFT	Bit position (2) for Lane 0 pre-emphasis in DP_SRC_ANALOG register
#define DPSRC_ANALOG_PREE_L0_MASK	Bit mask (0x3) for 2-bit Lane 0 pre-emphasis field
#define DPSRC_ANALOG_VOD_L1_SHIFT	Bit position (4) for Lane 1 VOD in DP_SRC_ANALOG register
#define DPSRC_ANALOG_VOD_L1_MASK	Bit mask (0x3) for 2-bit Lane 1 VOD field
#define DPSRC_ANALOG_PREE_L1_SHIFT	Bit position (6) for Lane 1 pre-emphasis in DP_SRC_ANALOG register
#define DPSRC_ANALOG_PREE_L1_MASK	Bit mask (0x3) for 2-bit Lane 1 pre-emphasis field
#define DPSRC_ANALOG_VOD_L2_SHIFT	Bit position (8) for Lane 2 VOD in DP_SRC_ANALOG register
#define DPSRC_ANALOG_VOD_L2_MASK	Bit mask (0x3) for 2-bit Lane 2 VOD field
#define DPSRC_ANALOG_PREE_L2_SHIFT	Bit position (10) for Lane 2 pre-emphasis in DP_SRC_ANALOG register
#define DPSRC_ANALOG_PREE_L2_MASK	Bit mask (0x3) for 2-bit Lane 2 pre-emphasis field
#define DPSRC_ANALOG_VOD_L3_SHIFT	Bit position (12) for Lane 3 VOD in DP_SRC_ANALOG register
#define DPSRC_ANALOG_VOD_L3_MASK	Bit mask (0x3) for 2-bit Lane 3 VOD field
#define DPSRC_ANALOG_PREE_L3_SHIFT	Bit position (14) for Lane 3 pre-emphasis in DP_SRC_ANALOG register
#define DPSRC_ANALOG_PREE_L3_MASK	Bit mask (0x3) for 2-bit Lane 3 pre-emphasis field
#define DPSRC_HPD_EVENT_SHIFT	Bit position (0) for HPD event type in DP_SRC_HPD_EVENT register
#define DPSRC_HPD_EVENT_MASK	Bit mask (0x3) for 2-bit HPD event type field
#define DPSRC_HPD_STS_SHIFT	Bit position (2) for current HPD pin status in DP_SRC_HPD_EVENT register
#define DPSRC_HPD_STS_MASK	Bit mask (0x1) for single-bit HPD pin status flag
#define DPSRC_PHY_RECONFIG_LINKRATE_SHIFT	Bit position (0) for PHY reconfiguration trigger in DP_SRC_PHY_RECONFIG register
#define DPSRC_PHY_RECONFIG_LINKRATE_MASK	Bit mask (0x1) for single-bit PHY reconfiguration trigger
#define DPSRC_PHY_READY_SHIFT	Bit position (31) for PHY ready status in DP_SRC_PHY_RECONFIG register

Macro	Description
#define DPSRC_PHY_READY_MASK	Bit mask (0x1) for single-bit PHY ready status flag
#define DPSRC_MSA0_MVID_SHIFT	Bit position (0) for MVID (M value) in DP_SRC_MSA0 register
#define DPSRC_MSA0_MVID_MASK	Bit mask (0xFFFFFF) for 24-bit MVID field
#define DPSRC_MSA1_NVID_SHIFT	Bit position (0) for NVID (N value) in DP_SRC_MSA1 register
#define DPSRC_MSA1_NVID_MASK	Bit mask (0xFFFFFF) for 24-bit NVID field
#define DPSRC_MSA2_HTOTAL_SHIFT	Bit position (0) for horizontal total in DP_SRC_MSA2 register
#define DPSRC_MSA2_HTOTAL_MASK	Bit mask (0xFFFF) for 16-bit horizontal total field
#define DPSRC_MSA2_VTOTAL_SHIFT	Bit position (16) for vertical total in DP_SRC_MSA2 register
#define DPSRC_MSA2_VTOTAL_MASK	Bit mask (0xFFFF) for 16-bit vertical total field
#define DPSRC_MSA3_HWIDTH_SHIFT	Bit position (0) for horizontal active width in DP_SRC_MSA3 register
#define DPSRC_MSA3_HWIDTH_MASK	Bit mask (0xFFFF) for 16-bit horizontal active width field
#define DPSRC_MSA3_VHEIGHT_SHIFT	Bit position (16) for vertical active height in DP_SRC_MSA3 register
#define DPSRC_MSA3_VHEIGHT_MASK	Bit mask (0xFFFF) for 16-bit vertical active height field
#define DPSRC_MSA4_HSTART_SHIFT	Bit position (0) for horizontal blanking start position in DP_SRC_MSA4 register
#define DPSRC_MSA4_HSTART_MASK	Bit mask (0xFFFF) for 16-bit horizontal start position field
#define DPSRC_MSA4_VSTART_SHIFT	Bit position (16) for vertical blanking start position in DP_SRC_MSA4 register
#define DPSRC_MSA4_VSTART_MASK	Bit mask (0xFFFF) for 16-bit vertical start position field
#define DPSRC_MSA5_HSW_SHIFT	Bit position (0) for horizontal sync width LSB in DP_SRC_MSA5 register
#define DPSRC_MSA5_HSW_MASK	Bit mask (0x1) for 1-bit horizontal sync width LSB
#define DPSRC_MSA5_HSP_SHIFT	Bit position (1) for horizontal sync polarity and width MSBs in DP_SRC_MSA5 register
#define DPSRC_MSA5_HSP_MASK	Bit mask (0x7FFF) for 15-bit horizontal sync polarity and width field
#define DPSRC_MSA5_VSW_SHIFT	Bit position (16) for vertical sync width LSB in DP_SRC_MSA5 register
#define DPSRC_MSA5_VSW_MASK	Bit mask (0x1) for 1-bit vertical sync width LSB
#define DPSRC_MSA5_VSP_SHIFT	Bit position (17) for vertical sync polarity and width MSBs in DP_SRC_MSA5 register
#define DPSRC_MSA5_VSP_MASK	Bit mask (0x7FFF) for 15-bit vertical sync polarity and width field
#define DPSRC_MSA6_MISC0_SHIFT	Bit position (0) for miscellaneous attribute 0 in DP_SRC_MSA6 register
#define DPSRC_MSA6_MISC0_MASK	Bit mask (0xFF) for 8-bit MISC0 field
#define DPSRC_MSA6_MISC1_SHIFT	Bit position (8) for miscellaneous attribute 1 in DP_SRC_MSA6 register
#define DPSRC_MSA6_MISC1_MASK	Bit mask (0xFF) for 8-bit MISC1 field
#define DPSRC_VBID_MSA_STS_VBID_SHIFT	Bit position (0) for Vertical Blanking ID in DP_SRC_VBID_MSA_STS register
#define DPSRC_VBID_MSA_STS_VBID_MASK	Bit mask (0x7F) for 7-bit VBID field
#define DPSRC_VBID_MSA_STS_UPDATE_SHIFT	Bit position (31) for MSA update trigger in DP_SRC_VBID_MSA_STS register
#define DPSRC_VBID_MSA_STS_UPDATE_MASK	Bit mask (0x1) for single-bit MSA update trigger flag
#define DPSRC_TU_CALC_VALID_BYTES_TU_SHIFT	Bit position (0) for valid bytes per transfer unit in DP_SRC_TU_CALC register
#define DPSRC_TU_CALC_VALID_BYTES_TU_MASK	Bit mask (0x7F) for 7-bit valid bytes per TU field
#define DPSRC_TU_CALC_VALID_BYTES_LANE_SHIFT	Bit position (8) for valid bytes per lane per TU in DP_SRC_TU_CALC register
#define DPSRC_TU_CALC_VALID_BYTES_LANE_MASK	Bit mask (0xFFFFFF) for 24-bit valid bytes per lane field
#define DPSINK_CTRL_LINK_RATE_CAP_SHIFT	Bit position (0) for maximum link rate capability in DP_SINK_CTRL register

Macro	Description
#define DPSINK_CTRL_LINK_RATE_CAP_MASK	Bit mask (0xFF) for 8-bit link rate capability field
#define DPSINK_CTRL_LANE_CNT_CAP_SHIFT	Bit position (8) for maximum lane count capability in DP_SINK_CTRL register
#define DPSINK_CTRL_LANE_CNT_CAP_MASK	Bit mask (0x7) for 3-bit lane count capability field
#define DPSINK_CTRL_CORE_RESET_SHIFT	Bit position (24) for core reset control in DP_SINK_CTRL register
#define DPSINK_CTRL_CORE_RESET_MASK	Bit mask (0x1) for single-bit core reset control flag
#define DPSINK_LT_SEQ_LINK_TRAIN_SEQ_SHIFT	Bit position (0) for link training sequence state in DP_SINK_LT_SEQ register
#define DPSINK_LT_SEQ_LINK_TRAIN_SEQ_MASK	Bit mask (0x7) for 3-bit training sequence state field
#define DPSINK_LT_SEQ_SCRAMBLE_DIS_SHIFT	Bit position (3) for scrambler disabled status in DP_SINK_LT_SEQ register
#define DPSINK_LT_SEQ_SCRAMBLE_DIS_MASK	Bit mask (0x1) for single-bit scrambler disabled status flag
#define DPSINK_LT_SEQ_LANE_CNT_SHIFT	Bit position (4) for configured lane count in DP_SINK_LT_SEQ register
#define DPSINK_LT_SEQ_LANE_CNT_MASK	Bit mask (0x7) for 3-bit lane count field
#define DPSINK_LT_SEQ_ENHANCE_FRAME_EN_SHIFT	Bit position (7) for enhanced framing enabled status in DP_SINK_LT_SEQ register
#define DPSINK_LT_SEQ_ENHANCE_FRAME_EN_MASK	Bit mask (0x1) for single-bit enhanced framing status flag
#define DPSINK_LT_SEQ_LINK_BW_SHIFT	Bit position (8) for configured link bandwidth in DP_SINK_LT_SEQ register
#define DPSINK_LT_SEQ_LINK_BW_MASK	Bit mask (0xFF) for 8-bit link bandwidth field
#define DPSINK_LT_SEQ_PHY_READY_SHIFT	Bit position (31) for PHY ready status in DP_SINK_LT_SEQ register
#define DPSINK_LT_SEQ_PHY_READY_MASK	Bit mask (0x1) for single-bit PHY ready status flag
#define DPSINK_LINK_CR_DONE_SHIFT	Bit position (0) for clock recovery done status in DP_SINK_LINK register
#define DPSINK_LINK_CR_DONE_MASK	Bit mask (0xF) for 4-bit clock recovery status field (one bit per lane)
#define DPSINK_LINK_SYM_LOCK_SHIFT	Bit position (4) for symbol lock status in DP_SINK_LINK register
#define DPSINK_LINK_SYM_LOCK_MASK	Bit mask (0xFF) for 8-bit symbol lock status field (two bits per lane)
#define DPSINK_LINK_INTERLANE_ALIGN_SHIFT	Bit position (12) for inter-lane alignment status in DP_SINK_LINK register
#define DPSINK_LINK_INTERLANE_ALIGN_MASK	Bit mask (0x1) for single-bit inter-lane alignment status flag
#define DPSINK_HPD_OVERRIDE_SHIFT	Bit position (0) for HPD override enable in DP_SINK_HPD_EVENT register
#define DPSINK_HPD_OVERRIDE_MASK	Bit mask (0x1) for single-bit HPD override enable flag
#define DPSINK_HPD_FORCE_SHIFT	Bit position (1) for forced HPD value in DP_SINK_HPD_EVENT register
#define DPSINK_HPD_FORCE_MASK	Bit mask (0x1) for single-bit forced HPD value flag
#define DPSINK_HPD_IRQ_EN_SHIFT	Bit position (2) for HPD interrupt enable in DP_SINK_HPD_EVENT register
#define DPSINK_HPD_IRQ_EN_MASK	Bit mask (0x1) for single-bit HPD interrupt enable flag
#define DPSINK_MSA0_MVID_SHIFT	Bit position (0) for received MVID in DP_SINK_MSA0 register
#define DPSINK_MSA0_MVID_MASK	Bit mask (0xFFFFFF) for 24-bit received MVID field
#define DPSINK_MSA1_NVID_SHIFT	Bit position (0) for received NVID in DP_SINK_MSA1 register
#define DPSINK_MSA1_NVID_MASK	Bit mask (0xFFFFFF) for 24-bit received NVID field
#define DPSINK_MSA2_HTOTAL_SHIFT	Bit position (0) for received horizontal total in DP_SINK_MSA2 register
#define DPSINK_MSA2_HTOTAL_MASK	Bit mask (0xFFFF) for 16-bit received horizontal total field
#define DPSINK_MSA2_VTOTAL_SHIFT	Bit position (16) for received vertical total in DP_SINK_MSA2 register
#define DPSINK_MSA2_VTOTAL_MASK	Bit mask (0xFFFF) for 16-bit received vertical total field
#define DPSINK_MSA3_HWIDTH_SHIFT	Bit position (0) for received horizontal active width in DP_SINK_MSA3 register
#define DPSINK_MSA3_HWIDTH_MASK	Bit mask (0xFFFF) for 16-bit received horizontal width field

Macro	Description
#define DPSINK_MSA3_VHEIGHT_SHIFT	Bit position (16) for received vertical active height in DP_SINK_MSA3 register
#define DPSINK_MSA3_VHEIGHT_MASK	Bit mask (0xFFFF) for 16-bit received vertical height field
#define DPSINK_MSA4_HSTART_SHIFT	Bit position (0) for received horizontal blanking start in DP_SINK_MSA4 register
#define DPSINK_MSA4_HSTART_MASK	Bit mask (0xFFFF) for 16-bit received horizontal start field
#define DPSINK_MSA4_VSTART_SHIFT	Bit position (16) for received vertical blanking start in DP_SINK_MSA4 register
#define DPSINK_MSA4_VSTART_MASK	Bit mask (0xFFFF) for 16-bit received vertical start field
#define DPSINK_MSA5_HSW_SHIFT	Bit position (0) for received horizontal sync width LSB in DP_SINK_MSA5 register
#define DPSINK_MSA5_HSW_MASK	Bit mask (0x1) for 1-bit received horizontal sync width LSB
#define DPSINK_MSA5_HSP_SHIFT	Bit position (1) for received horizontal sync polarity and width in DP_SINK_MSA5 register
#define DPSINK_MSA5_HSP_MASK	Bit mask (0x7FFF) for 15-bit received horizontal sync polarity and width field
#define DPSINK_MSA5_VSW_SHIFT	Bit position (16) for received vertical sync width LSB in DP_SINK_MSA5 register
#define DPSINK_MSA5_VSW_MASK	Bit mask (0x1) for 1-bit received vertical sync width LSB
#define DPSINK_MSA5_VSP_SHIFT	Bit position (17) for received vertical sync polarity and width in DP_SINK_MSA5 register
#define DPSINK_MSA5_VSP_MASK	Bit mask (0x7FFF) for 15-bit received vertical sync polarity and width field
#define DPSINK_MSA6_MISC0_SHIFT	Bit position (0) for received miscellaneous attribute 0 in DP_SINK_MSA6 register
#define DPSINK_MSA6_MISC0_MASK	Bit mask (0xFF) for 8-bit received MISC0 field
#define DPSINK_MSA6_MISC1_SHIFT	Bit position (8) for received miscellaneous attribute 1 in DP_SINK_MSA6 register
#define DPSINK_MSA6_MISC1_MASK	Bit mask (0xFF) for 8-bit received MISC1 field
#define DPSINK_VBID_MSA_STS_VBID_SHIFT	Bit position (0) for received Vertical Blanking ID in DP_SINK_VBID_MSA_STS register
#define DPSINK_VBID_MSA_STS_VBID_MASK	Bit mask (0x3F) for 6-bit received VBID field
#define DPSINK_VBID_MSA_STS_LOCK_SHIFT	Bit position (7) for MSA lock status in DP_SINK_VBID_MSA_STS register
#define DPSINK_VBID_MSA_STS_LOCK_MASK	Bit mask (0x1) for single-bit MSA lock status flag
#define DPSINK_VBID_MSA_STS_UPDATED_SHIFT	Bit position (8) for MSA updated flag in DP_SINK_VBID_MSA_STS register
#define DPSINK_VBID_MSA_STS_UPDATED_MASK	Bit mask (0x1) for single-bit MSA updated flag
#define DPSINK_BER_SEL_SHIFT	Bit position of BER lane select (bit 0; 0 = lane 0–1, 1 = lane 2–3)
#define DPSINK_BER_SEL_MASK	1-bit mask for BER lane select
#define DPSINK_BER_CLR_SHIFT	Bit position of BER counter clear (bit 1; write-1 to reset counters)
#define DPSINK_BER_CLR_MASK	1-bit mask for BER counter clear
#define DPSINK_BER0_LANE0_SHIFT	Bit position of BER error count for lane 0 (bit [15:0])
#define DPSINK_BER0_LANE0_MASK	16-bit mask for BER error count for lane 0
#define DPSINK_BER0_LANE1_SHIFT	Bit position of BER error count for lane 1 (bit [31:16])
#define DPSINK_BER0_LANE1_MASK	16-bit mask for BER error count for lane 1
#define DPSINK_BER1_LANE2_SHIFT	Bit position of BER error count for lane 2 (bit [15:0])
#define DPSINK_BER1_LANE2_MASK	16-bit mask for BER error count for lane 2
#define DPSINK_BER1_LANE3_SHIFT	Bit position of BER error count for lane 3 (bit [31:16])
#define DPSINK_BER1_LANE3_MASK	16-bit mask for BER error count for lane 3

Macro	Description
#define REG_GET_FIELD	Extracts a multi-bit field: right-shift reg by shift, then AND with mask
#define REG_SET_FIELD	Inserts val into a multi-bit field: clear existing bits, then OR in the new value
#define REG_GET_BIT	Reads a single bit: returns 0 or 1
#define REG_SET_BIT	Sets a single bit to 1
#define REG_CLEAR_BIT	Clears a single bit to 0
#define REG_TOGGLE_BIT	Toggles a single bit
#define REG_TEST_BIT	Tests a single bit; evaluates to true (non-zero) if set
#define DP_BASE_CLOCK_HZ	DisplayPort base clock frequency (270 MHz)
#define SYMBOL_PER_CLOCK	Number of symbols per clock cycle (10-bit encoding over 4 lanes)
#define COLOR_PLANE_MISC1_BIT_LEN	Color plane misc1 bit field length
#define COLOR_PLANE_MISC1_BIT_POS	Color plane misc1 bit field position
#define COLOR_PLANE_MISC0_BIT_LEN	Color plane misc0 bit field length
#define COLOR_PLANE_MISC0_BIT_POS	Color plane misc0 bit field position
#define COLOR_PLANE_SHIFT_AMOUNT	Shift amount for color plane index calculation
#define REG_SHIFT_16_BITS	16-bit shift for register packing
#define REG_SHIFT_15_BITS	15-bit shift for sync polarity
#define REG_SHIFT_31_BITS	31-bit shift for vertical sync polarity
#define REG_SHIFT_8_BITS	8-bit shift for TU calculation register
#define REG_SHIFT_4_BITS	4-bit shift for lane count extraction
#define REG_SHIFT_7_BITS	7-bit shift for enhanced framing
#define REG_SHIFT_3_BITS	3-bit shift for scrambler disable
#define FRAME_RATE_MULTIPLIER	Multiplier for frame rate calculation
#define BOOLEAN_FALSE	Boolean false value
#define BOOLEAN_TRUE	Boolean true value
#define INITIALIZED_FLAG	Handle initialization flag value
#define LANE_COUNT_1	Single lane configuration
#define LANE_COUNT_2	Dual lane configuration
#define LANE_COUNT_3	Quad lane configuration

References

- DisplayPort IP User Guide (FPGA-IPUG-02236)
- DisplayPort IP Release Notes (FPGA-RN-02071)
- CertusPro-NX web page
- Certus-N2 web page
- Avant-G web page
- Avant-X web page
- DisplayPort IP Core web page
- Lattice Radiant Software web page
- Lattice Propel Design Environment web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.1, June 2026

Section	Change Summary
All	Performed minor formatting and editorial edits.
Abbreviations in This Document	Updated list of abbreviations.
Introduction	• Added Lattice Avant and Certus-N2 devices in the Audience section. • Updated driver version in the Driver Version section. • Updated driver version and IP version in Table 1.1. Driver and IP Compatibility.

API Description	• Updated parameter from base_addr to base_address in the dp_sink_init() section.
	• Updated API code in the following sections:
	  • dp_sink_get_msa_lock()
	  • dp_sink_get_msa_updated()
	• Updated the return for the handle parameter in the dp_sink_read_config() section.
	• Updated the dp_src_init() section as follows:
	  • Updated API code.
	  • Updated description for the handle parameter.
	  • Updated parameter from base_addr to base_address.
	• Updated the dp_src_msa_set() section as follows:
	  • Updated API description.
	  • Updated API code.
	  • Updated parameter from handle to src_handle.
	• Updated the return for the handle parameter in the dp_src_read_config() section.
	• Removed the *dp_src_cmd()* section.

Function Call Flow Diagrams	• Updated the following figures:
	  • Figure 3.1. unsigned char dp_sink_init()
	  • Figure 3.2. unsigned char dp_sink_cmd()
	  • Figure 3.3. unsigned char dp_sink_get_msa_lock()
	  • Figure 3.5. dp_config_t* dp_sink_read_config()
	  • Figure 3.6. unsigned char dp_src_init()
	  • Figure 3.7. unsigned char dp_src_msa_set()
	  • Figure 3.8. dp_config_t* dp_src_read_config()
	• Removed the *dp_src_cmd()* section.

API Data Structures	• Removed the *struct dp_handle_t* section.
	• Added DP_SRC_ANALOG_REG in Table 4.1. src_reg_map_t Parameters.
	• Updated DP_SINK_LT_SEQ_STS_REG to DP_LT_SEQ_STS_REG, DP_SINK_LINK_STS_REG to DP_LINK_STS_REG, and DP_SINK_HPD_EVENT_REG to DP_HPD_EVENT_REG in Table 4.2. sink_reg_map_t Parameters.
	• Updated Table 4.3. dp_video_timing_t Parameters as follows:
	  • Updated data type.
	  • Added h_start and h_start.
	• Updated Table 4.4. dp_config_t Parameters as follows:
	  • Added msa6_misc.
	  • Updated data type for enhanced_framing and scrambler_disable.
	• Updated data type in Table 4.5. dp_status_t Parameters.
	• Updated data type in Table 4.6. dp_src_handle_t Parameters.
	• Renamed table from *dp_status_t Parameters* to Table 4.7. dp_sink_handle_t Parameters and updated data type.

API Enum	• Updated Table 5.1. dp_src_cmd_st Variables.
	• Updated section and table titles in the following sections:
	  • enum dp_link_rate_t
	  • enum dp_lane_count_t

Section	Change Summary

	• enum dp_color_depth_t
	• enum dp_color_format_t
	• Updated Table 5.6. dp_sink_cmd_st Variables as follows:
	  • Updated DP_SINK_SET_LINK_RATE_CAP to DP_SINK_SET_LINK_RATE and DP_SINK_SET_LINK_CNT_CAP to DP_SINK_SET_LINK_CNT.
	  • Updated the value for DP_SINK_CMD_MAX.

API Macros	Updated Table 6.1. API Macros.
References	Updated references.

Revision 1.0, December 2025

Section	Change Summary
All	Initial release.

www.latticesemi.com