I3C Filter Driver API Reference

Technical Note

FPGA-TN-02390-1.2

December 2025

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

Figures

Tables

Abbreviations in This Document

A list of acronyms used in this document.

Abbreviations	Definition
API	Application Programming Interface
FPGA	Field Programmable Gate Array
I/O	Input/Output
I2C	Inter-Integrated Circuit
I3C	Improved Inter-Integrated Circuit
ID	Identification
IP	Intellectual Property
SMBus	System Management Bus

1. Introduction

The I3C Filter IP functions as an invisible relay from the point of view of the controller and target devices on the bus. The IP must be connected between the controller and all target devices at the SDA and SCL ports. This IP protects all target devices from malicious traffic generated from the controller port based on a list of allowable commands set by the host. The I3C Filter IP supports the I3C protocol. The IP can also be configured as an I2C Filter. In the I2C Filter mode, the IP is a subset of the SMBus protocol.

Refer to the I3C Filter IP User Guide (FPGA-IPUG-02257) 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 the Lattice MachXO5™-NX devices. The technical guide assumes readers have expertise in embedded systems and FPGA technologies.

1.3. Driver Version

I3C_FILTER_DRV_VER *v25.2.0*.

1.4. Driver and IP Compatibility

Driver Version	IP Version
25.2.0	1.7.0

Refer to the I3C Filter IP Release Notes (FPGA-RN-02004) for more information on the driver and IP versions.

2. API Description

2.1. smbus_filter_init()

This API initializes the I3C Filter IP with the following actions:

- Assigns the base address to the this_smbus_filter struct.
- Initializes the I3C or I2C device support based on the target address.
- Enables the interrupt (Offset 0x1014).

```
unsigned char smbus_filter_init(struct smbus_filter_instance *this_smbus_filter, unsigned int base_addr)
```

In/Out	Parameter	Description	Returns
In	this_smbus_filter	Handle of the smbus_filter_instance structure.	0: Success 1: Failure
In	base_addr	Base address to be assigned.	

2.2. enable_2byte_filter()

This API is used to set 1byte0_2byte1 (bit 0) in the filter configuration register (0x1000).

```
void enable_2byte_filter(unsigned int base_address, unsigned int enable_1byte_2byte)
```

In/Out	Parameter	Description	Returns
In	base_address	Base address to be assigned.	None
In	enable_1byte_2byte	Option to enable either 1-byte or 2-byte filter.	

2.3. i2c_device_info_back_ward_cmptibty()

This API is used to set the mapping of I3C or I2C device support based on the target address. The I3C Filter IP supports 128 target addresses and it is applicable only when I3C Filter is selected.

```
unsigned char i2c_device_info_back_ward_cmptibty(struct smbus_filter_instance *this_smbus_filter, unsigned int *i2c_device_info_word)
```

In/Out	Parameter	Description	Returns
In	this_smbus_filter	Handle of the smbus_filter_instance structure.	0: Success 1: Failure
In	i2c_device_info_word	Pointer to the array of target device type (I2C/I3C)	

2.4. init_tgt_dev()

This API functions as a wrapper for mapping I3C or I2C device support based on the Target Address. The I3C Filter IP supports 128 target addresses and it is applicable only when I3C Filter is selected.

```
unsigned char init_tgt_dev(volatile struct smbus_filter *this_smbus_filter, struct smbus_filter_instance *this_smbus_inst)
```

In/Out	Parameter	Description	Returns
In	this_smbus_filter	Handle of the smbus_filter structure.	None
In	this_smbus_inst	Handle of the smbus_filter_instance structure.	

2.5. assign_allow_list_to_target()

This API is used to assign allow list to the target address.

```
unsigned char assign_allow_list_to_target(unsigned int base_addr, unsigned int id, unsigned int target_addr)
```

In/Out	Parameter	Description	Returns
In	base_address	Base address to be assigned.	0: Success 1: Failure
In	id	ID of the allow list.	
In	target_addr	Target device address to be assigned.	

2.6. init_cmd_to_allow_list()

This API is used to initialize the allow list by mapping a 256-bit value for byte 1 and byte 2 allow list ranging from 0 to 59.

```
unsigned char init_cmd_to_allow_list(struct smbus_filter_manifest *sm_filter_manifest, unsigned int base_addr, unsigned int list_id)
```

In/Out	Parameter	Description	Returns
In	sm_filter_manifest	Handle of the smbus_filter_manifest structure.	0: Success 1: Failure
In	base_addr	Base address to be assigned.	
In	list_id	ID of the allow list.	

2.7. smbus_filter_set_whitelist()

This API is used to initialize the allow list and assign it to the target device accordingly.

```
unsigned char smbus_filter_set_whitelist(struct smbus_filter_manifest *sm_filter_manifest, struct smbus_filter_instance *this_smbus_filter, unsigned char list_id, unsigned char avoid_new_whitelist)
```

In/Out	Parameter	Description	Returns
In	sm_filter_manifest	Handle of the smbus_filter_manifest structure.	0: Success 1: Failure
In	this_smbus_filter	Handle of the smbus_filter_instance structure.	
In	list_id	ID of the allow list.	
In	avoid_new_whitelist	Indicates a new list to be initialized.	

2.8. smbus_filter_event_get()

This API is used to read the interrupt information stored in offset 0x1020.

```
unsigned char smbus_filter_event_get(struct smbus_filter_instance *this_filter, unsigned int *val)
```

In/Out	Parameter	Description	Returns
In	this_filter	Handle of the smbus_filter_manifest structure.	0: Success 1: Failure
In	val	Pointer to a variable for storing data retrieved from registers.	

2.9. smbus_filter_dis()

This API is used to controls the state of the SMBus filter for the specified filter instance.

```
unsigned char smbus_filter_dis(struct smbus_filter_instance *this_filter, unsigned char en)
```

In/Out	Parameter	Description	Returns
In	this_filter	Handle of the smbus_filter_manifest structure.	0: Success 1: Failure
In	en	Non-zero value to enable the filter, zero to disable	

3. Function Call Flow Diagrams

3.1. unsigned char smbus_filter_init()

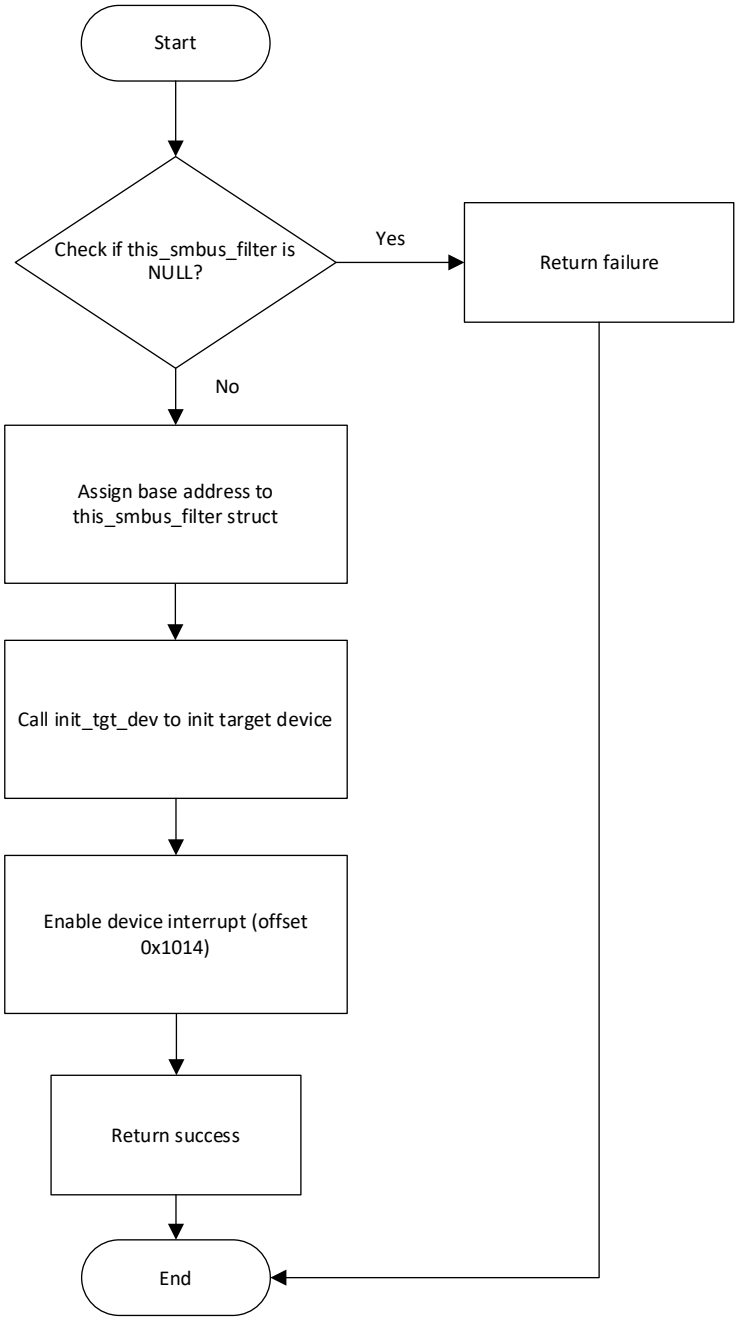

Figure 3.1. unsigned char smbus_filter_init

3.2. enable_2byte_filter()

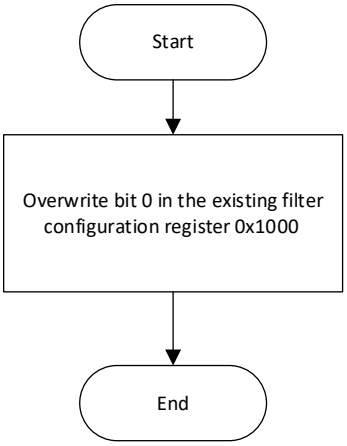

Figure 3.2. void enable_2byte_filter

3.3. i2c_device_info_back_ward_cmptibty()

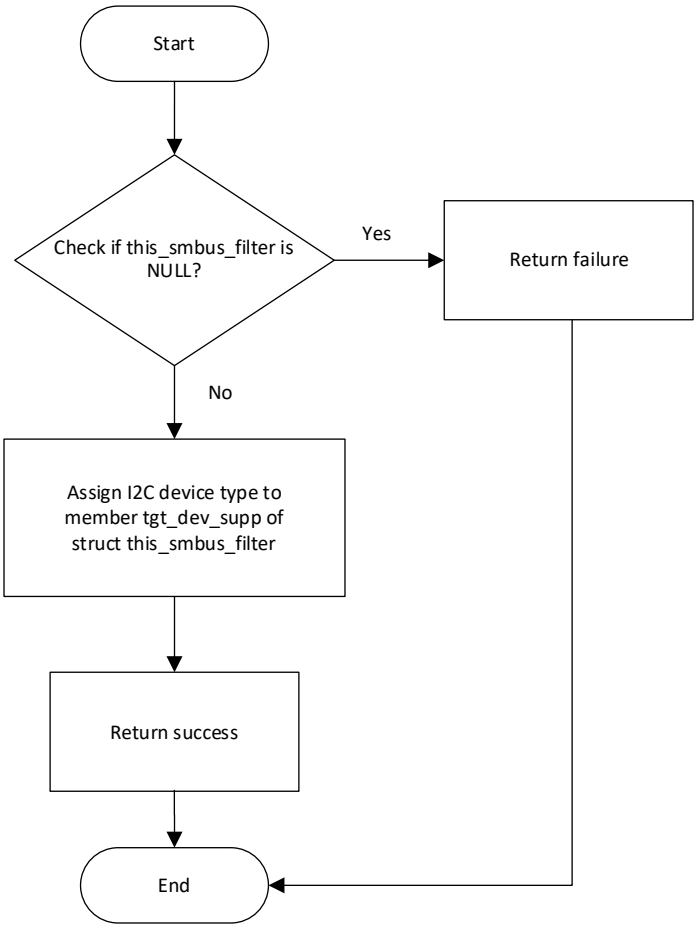

Figure 3.3. unsigned char i2c_device_info_back_ward_cmptibty

3.4. init_tgt_dev()

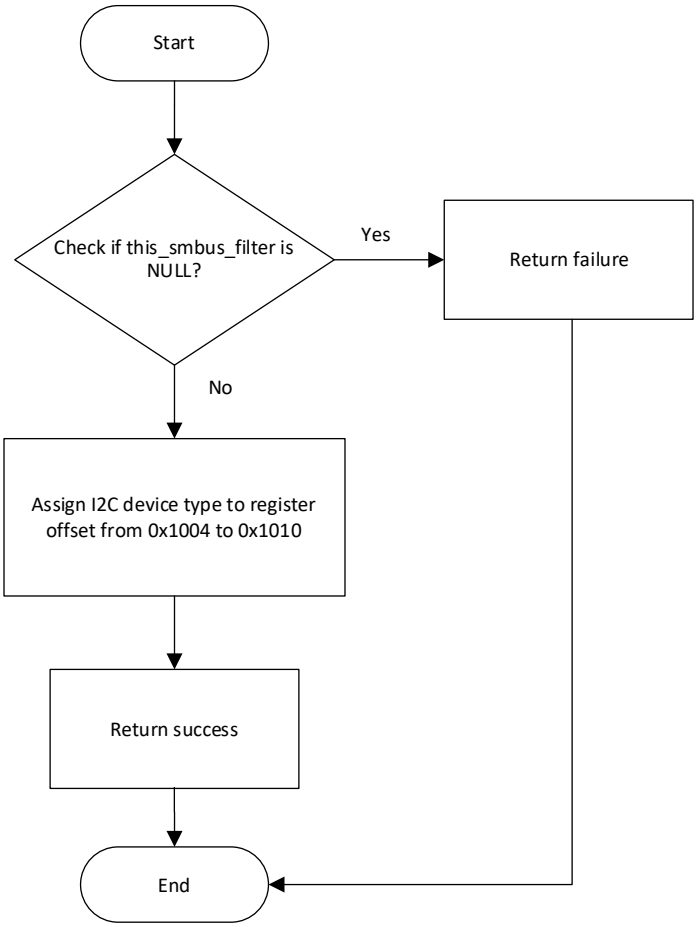

Figure 3.4. unsigned char init_tgt_dev

3.5. assign_allow_list_to_target()

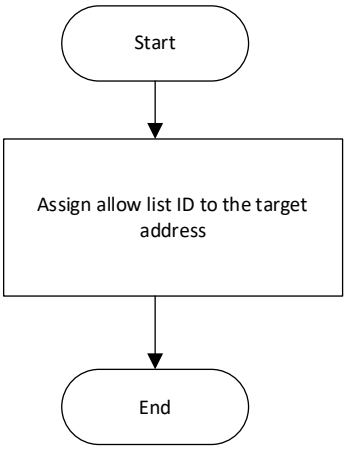

Figure 3.5. unsigned char assign_allow_list_to_target

3.6. init_cmd_to_allow_list()

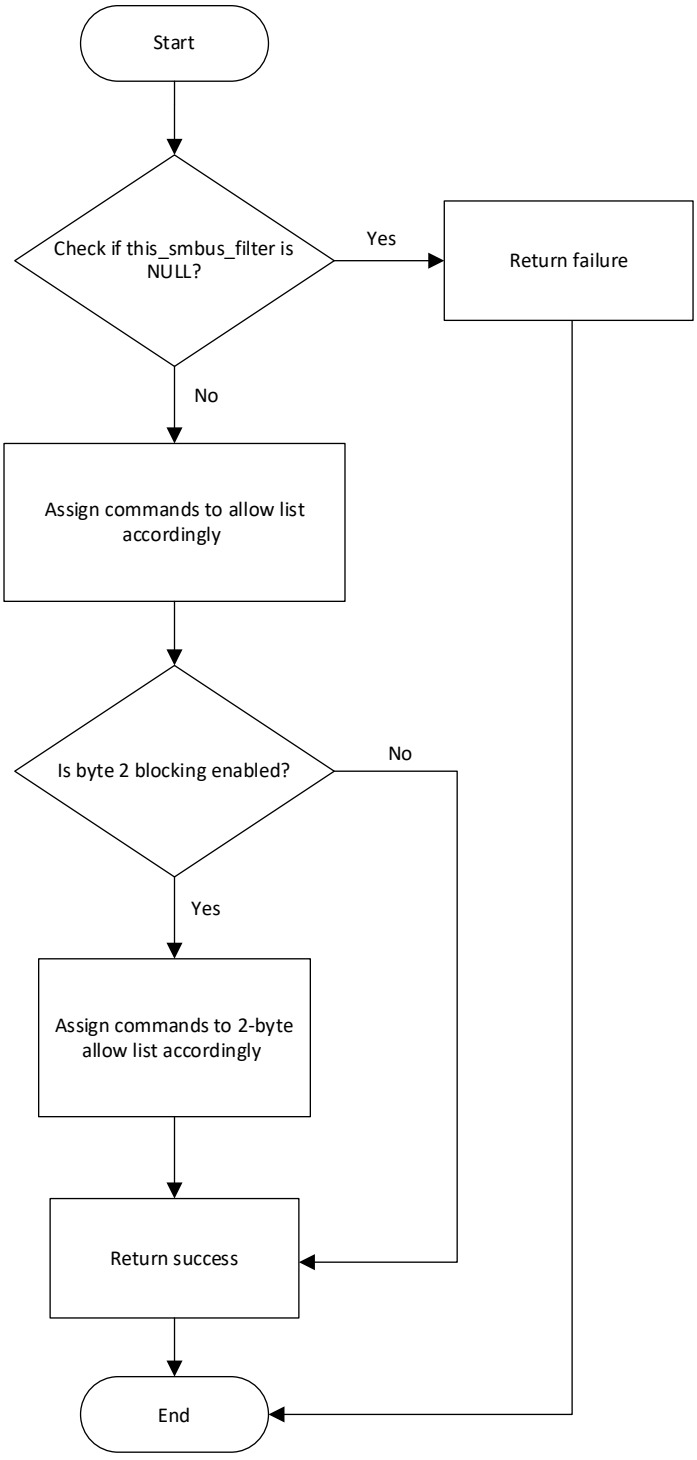

Figure 3.6. unsigned char init_cmd_to_allow_list

3.7. smbus_filter_set_whitelist()

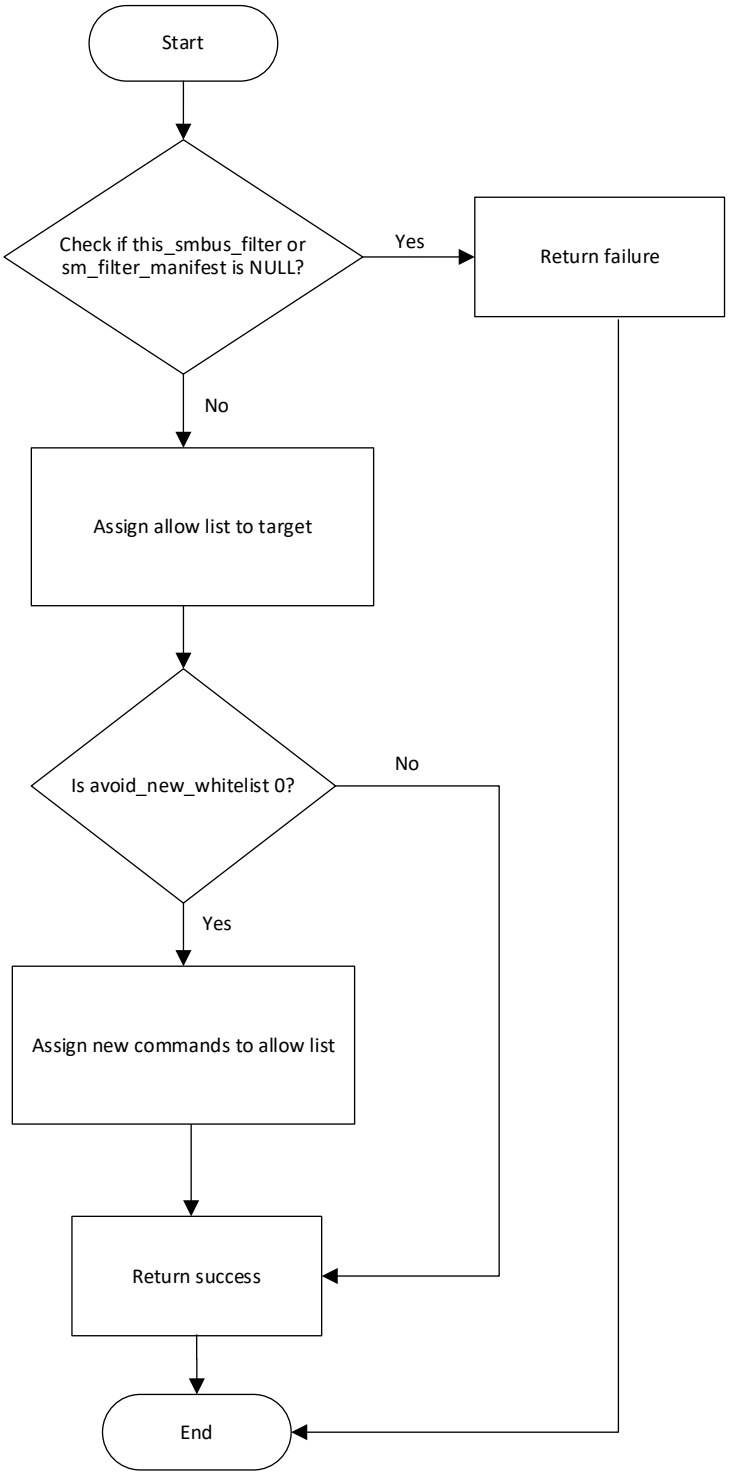

Figure 3.7. unsigned char smbus_filter_set_whitelist

3.8. smbus_filter_event_get()

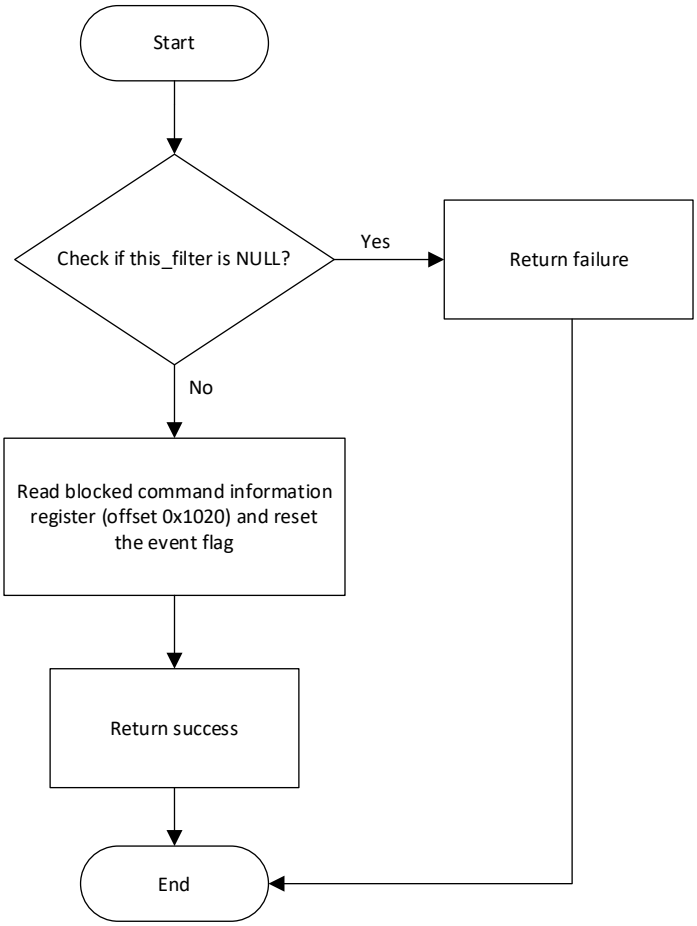

Figure 3.8. unsigned char smbus_filter_event_get

3.9. smbus_filter_dis()

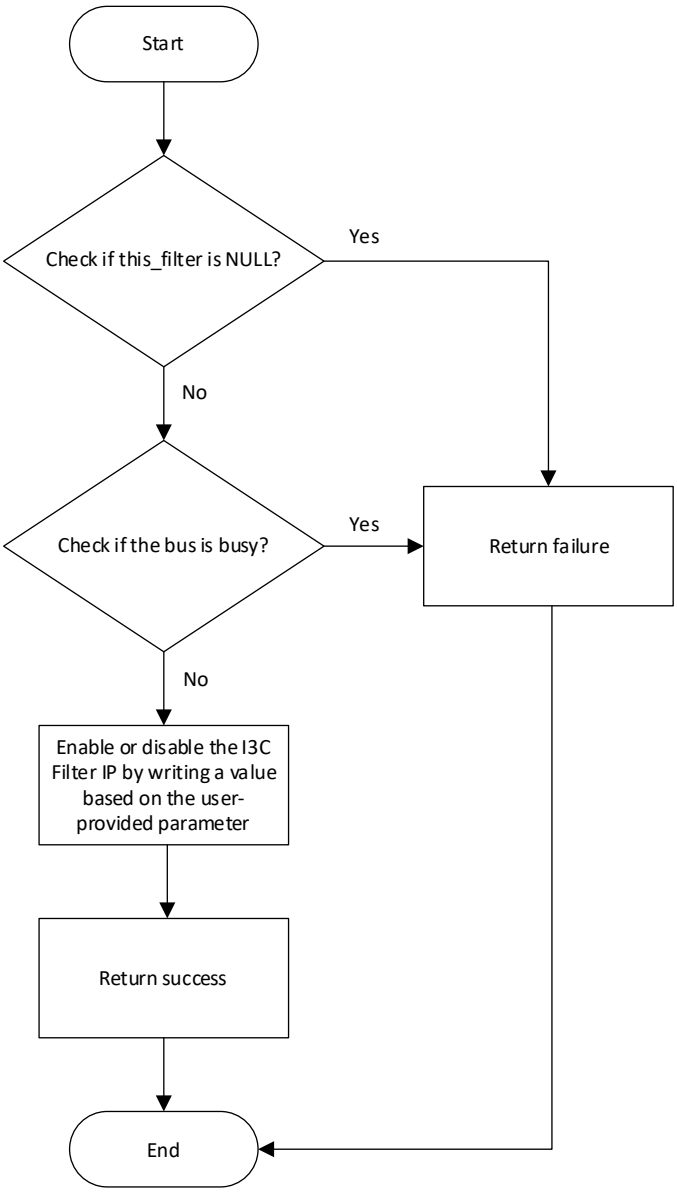

Figure 3.9. unsigned char smbus_filter_dis

4. API Data Structures

4.1. struct smbus_filter_instance

Table 4.1. smbus_filter_instance Parameters

Data Type	Struct Member	Description
unsigned int	base_address	I3C Filter IP base address.
unsigned char	event_flag	Triggered event indication.
unsigned int	whitelist_id[32]	Allow list array.
unsigned int	tgt_dev_supp[4]	Supported device types array (I2C/I3C).

4.2. struct smbus_filter

Table 4.2. smbus_filter Parameters

Data Type	Struct Member	Description
unsigned int	filter_config	Register of filter configuration (offset 0x1000).
unsigned int	tgt_dev_supp_1	Register of Target Device Support 1 (offset 0x1004).
unsigned int	tgt_dev_supp_2	Register of Target Device Support 2 (offset 0x1008).
unsigned int	tgt_dev_supp_3	Register of Target Device Support 3 (offset 0x100C).
unsigned int	tgt_dev_supp_4	Register of Target Device Support 4 (offset 0x1010).
unsigned int	irq_enable	Register of Interrupt Enable (offset 0x1014).
unsigned int	irq_status	Register of Interrupt Status (offset 0x1018).
unsigned int	irq_set	Register of Interrupt Set (offset 0x101C).
unsigned int	blk_cmd_info	Register of Blocked Command Information (offset 0x1020).
unsigned int	bus_status	Register of the I2C/I3C bus status.

4.3. struct smbus_filter_manifest

Table 4.3. smbus_filter_manifest Parameters

Data Type	Struct Member	Description
unsigned int	smbus_device_addr	Target device to be assigned for allow list.
unsigned char	smbus_cmd_passlist[NO_OF_DATA_BLOCKS]	Array to store the allowed commands.
unsigned int	smbus_cmd_passlist_2[NO_OF_DATA_BLOCKS]	Array to store the allowed commands (2-byte filter mode enabled).

5. API Macros

Table 5.1. API Macros Description

Macro	Description
#define WHITELIST_1ST_BYTE_FILTER_OFFSET	Offset of the allow list register.
#define WHITELIST_SIZE	Size of the allow list array.
#define NO_OF_DATA_BLOCKS	Block size of the allow list.
#define WHITELIST_2ND_BYTE_FILTER_OFFSET	Offset of the allow list filter list (2-byte filter mode) register.
#define RET_FAILURE	Return value of failure status.
#define RET_SUCCESS	Return value of success status.
#define CONFIG_FILTER_REG	Offset of the filter configuration register.
#define INT_EN_REG	Offset of the interrupt enable register.
#define INT_STS_REG	Offset of the interrupt status register.
#define INT_SET_REG	Offset of the interrupt set register.
#define BLK_CMD_INFO_REG	Offset of the blocked command information register.
#define BYTE_2_BLOCKING_MASK	Mask bit to indicate 1-byte or 2-byte blocking mode.

References

- I3C Filter IP User Guide (FPGA-IPUG-02257)
- I3C Filter IP Release Notes (FPGA-RN-02004)
- I3C Filter IP Core web page
- MachXO5-NX web page
- Lattice Solutions IP Cores web page
- Lattice Solutions Reference Designs web page
- Lattice Propel Design Environment web page
- Lattice Radiant Software web page
- Lattice Insights web page 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 1.2, December 2025

Section	Change Summary
Introduction	• Updated the driver version in the Driver Version section. • Updated the driver and IP versions in the Driver and IP Compatibility section.
API Description	Added the smbus_filter_dis() section.
Function Call Flow Diagrams	Added the smbus_filter_dis() section.
API Data Structures	Added *bus_status* to Table 4.2. smbus_filter Parameters.
References	Added the *I3C Filter IP Core* and *Lattice Solutions Reference Designs* web pages.

Revision 1.1, February 2025

Section	Change Summary
Abbreviations in This Document	Updated this section.
Introduction	In the Driver and IP Compatibility section: • Updated the *IP Version*. • Added the *I3C Filter IP Release Notes (FPGA-RN-02004)* reference.
Function Call Flow Diagrams	Updated the unsigned char smbus_filter_init() section title.
References	Added *I3C Filter IP Release Notes (FPGA-RN-02004)*.

Revision 1.0, January 2025

Section	Change Summary
All	Initial release.

www.latticesemi.com